2009

Model-based Systems Engineering approach to US Navy towing and salvage fleet recapitalization

Whitcomb, Cliffard A.; Addington, Christopher; Hlavin, Wenonah; Southworth, George

MODEL-BASED SYSTEMS ENGINEERING APPROACH TO US NAVY TOWING AND SALVAGE FLEET RECAPITALIZATION

Authors

Dr. Clifford A. Whitcomb, CSEP*
Associate Professor of Systems Engineering

LT Christopher Addington, USN
Graduate Student

Wenonah Hlavin
Research Associate

LCDR George Southworth, USN
Graduate Student

NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA

ABSTRACT

The United States Navy owns four salvage ships and four towing ships that will reach the end of their 40-year life expectancy in 2019. The program manager for these vessels has a set of desirable performance requirements for a new ship class, T-ARS(X), which combines the capabilities from both the salvage and towing ship classes. The need to develop a recapitalization strategy based on either designing a new ship class based on these desirable requirements or purchasing commercial capabilities based on the salvage and towing community's needs is paramount.

The Department of Defense (DoD) has shifted defense planning from the specific service requirements generating system (RGS) acquisition to the Joint Capabilities Integration and Development System (JCIDS) approach that focuses on requirements generation based on customer need. This paper explores how to use systems architecture development principles in the context of model-based systems engineering (MBSE) to incorporate the capabilities needed for towing and salvage recapitalization into a cohesive framework for developing the T-ARS(X) requirement specification. The CORE design tool is used to implement the MBSE architecting process using the Naval Architecture Elements Reference Guide (NAERG) and standardized operational tasks to create Department of Defense Architecture Framework (DoDAF) v1.5 products from system models. The requirements generated from the architecture model are compared with the current, combined towing and salvage-capable commercial platforms for analysis. The methodology presented provides the towing and salvage community with the basis to perform a capabilities-based analysis of alternatives (AoA) for the T-ARS(X) recapitalization.

INTRODUCTION

The United States Navy has a number of considerations relating to the recapitalization of their ocean-towing and salvage ships that are in need of replacement within the next 10 to 20 years. The need for ARS and T-ATF recapitalization has been verified and alternative acquisition strategies, such as building a new platform or purchasing from the commercial market, are being entertained (Sperling & Keenan, 2006).

Alternative possibilities to meet the Navy's towing and salvage requirements include building a new ship, purchasing commercial platforms, or a combination of both. For the alternative investment strategy of purchasing commercial platforms, a contractor-owned contractor-operated (COCO) option has been demonstrated by CNA to be more cost effective, based on current towing and salvage requirements (Sperling & Keenan, 2006). In order to consider stakeholder needs in the decision-making process for such a future Fleet investment, a study of the architecture of the elements involved, and their relationships, should be conducted. The ocean-towing and salvage capability is a System of Systems (SoS) and the need to define an adequate architecture is key to identifying the top-level requirements that are crucial in determining which investment strategy the Navy should consider.

The motivation for this paper is to develop a capabilities-driven architecture development process, integrated into a model-based systems engineering methodology, and to demonstrate a thorough consideration of capabilities to develop the basis for an AoA that considers requirements for both new platform options as well as COCO assets from the commercial market. Key outcomes described in this paper are:

- An architecture and architecture-based requirements generation process (with focus on stakeholder needs) ideally suited to future salvage platform force structure development.
- A model-based systems engineering (MBSE) process that integrates architecting principles, from engineering requirements definition to physical architecture integration, for fusing the diverse assets involved in this complex system (Whitcomb, Vaidyanathan, et al, 2008).
- A set of architectural and realizable requirement specifications based on the salvage community's needs.
- An architecture based on capabilities mapped to mission activities.
- A brief market analysis of the COCO possibilities against requirements.

The end result is a method that enables effective decision-making efforts for the recapitalization of the "Future Salvage Platform Force Structure," as well as preventing the expenditure of resources in areas that may not be feasible in the period of development, thus ensuring a sound basis of architecture for the future salvage fleet (Whitcomb, Vaidyanathan, et al, 2008). The development of this model-based methodology requires consideration of many newly architectural aspects of systems – from systems engineering and architecting, capabilities-based planning, SoS, and architectural elements.

Systems Engineering

Systems engineering is generally used to describe the set of processes applied to the development of a system that consists of two significant disciplines: the technical knowledge domain in which the systems engineer operates, and systems engineering management (DAU, 2001). Systems engineering spans the progression from customer need discovery to the disposal of the system. A system is an integrated group which embodies a set of relationships among the composite of people, products, and processes, providing a capability to satisfy a stated need or objective (DAU, 2001).

Once the requirements have been identified and defined, they need to be mapped to functions, which must be analyzed and allocated into lower-level functions. Higher-level functions can be analyzed by decomposing them into lower-level functions. “The result is a description of the product or item in terms of what it does logically and in terms of the performance required” (DAU, 2001). Lower-level functions derived from higher-level functions provide a better understanding of what the actual functions are and how they are associated with each other. This description is called a functional architecture and provides “information essential to optimizing physical solutions” (DAU, 2001).

After all functions have been identified, each must then be matched to a requirement for use in developing the physical architecture initialization through design synthesis. Design synthesis is the process of defining the physical architecture of the system in terms of its physical elements. Each physical element must meet at least one functional requirement. “The physical architecture is the basic structure for generating the specifications and baselines” (DAU, 2001). During the physical architecture synthesis, it is consistently aligned with the functional architecture, eventually with physical system performance verified to the requirements in a design loop. The verification process is a formal testing and evaluation procedure for ensuring that all requirements will be met by the proposed solution.

Systems Architecture and Architecting

The early-stage activities involved in systems engineering have salient features more related to the field of architecture than that of engineering. The difference between architecting and engineering is described in terms of “art and science” (Maeir & Rectin, 2002). Architecting focuses on the architecture, or art, and patience of a designer necessary to complement the complexity of engineering the system. Architecting contrasts with engineering in that it is “nonanalytic, difficult to clarify, and seldom taught formally in industry” (Maeir & Rectin, 2002). Architecting plays a vital role in creating new types of complex systems that incorporate evolving technologies (Maeir & Rectin, 2002). The need for architecting is shown in that it complements engineering in accounting for the immeasurable: e.g., multiple stakeholders, perceptions of worth, safety, affordability, political acceptance, and environmental impact. Therefore, the development of an architecture in the earliest stages of a systems engineering process is justified.

System of Systems (SoS)

The concept of systems has been recently expanded to directly define “SoS” as unique from “systems.” An SoS is defined as “a set or arrangement of systems that results when independent and useful systems are integrated into a larger system that delivers unique capabilities” (DoD, 2005). The SoS concept does not specify a need for particular new methods; instead, it suggests a new way of thinking for solving complex interactions of technology, policy, and economics.

System of Systems (SoS) Architecting

The role of SoS architecting in the SE process is to integrate functional architecture within the functional analysis/allocation design loop. The architecting of an SoS starts with the transformation of an operational capability need into a set of requirements, which are used to guide the development of functional and physical architectures through design (Whitcomb, Vaidyanathan, et al, 2008). The development of functional architectures will bridge the gap between the stakeholders' needs and an understandable functional breakdown structure of the collected requirements.

Capabilities-Based Systems Development

The DoD is in the process of implementing a capabilities-based requirements-to-resources system. The DoD directed the initiation of a capabilities-based approach to defining defense requirements (Walker, 2005). The emphasis was placed on delivering capabilities to address a wide range of mission objectives of the future towing and salvage platform(s).

A major factor that could inhibit the future salvage platform from meeting its full potential is that the proposed top-level characteristics are requirements driven, with the initial designer having a preconceived notion of the solution. The method of developing systems, as well as the new systems development process, called the Joint Capabilities Integration Development System (JCIDS), is shown in Figure 1.

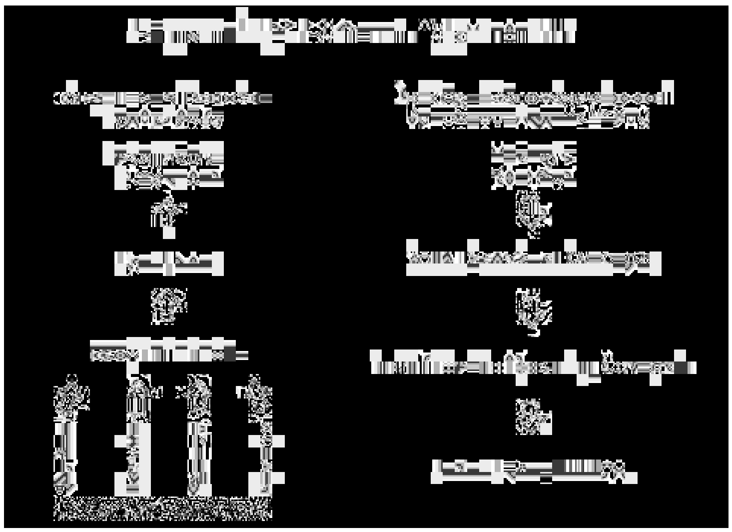

Figure 1. New Capabilities-Based Acquisition approach (Walker, 2005)

The left-hand side of Figure 1 represents a simplified version of the old Requirements Generation System (Walker, 2005). The old method concentrated on generating requirements in order to fulfill their "idea" of warfighting. These required capabilities were derived within a system where joint service contributions were ignored. The new capability-based planning approach is represented on the right-hand side of Figure 1. Instead of trying to generate interservice requirements, based on joint service capabilities, at the end of the process, the new approach inverts the paradigm, concentrating on the capabilities of the joint services at the beginning of the process (Walker, 2005). An architecting process can provide the capabilities needed by pulling the requirements based on mission need, and by focusing on the problem rather than the solution.

The JCIDS is one component of the capability-based planning (CBP) process that the DoD uses as its principal decision support process for transforming the military to support the national military and defense strategy. JCIDS plays a key role in identifying the capabilities required by the warfighters to support the national defense strategy and the national military strategy. The procedures established in the JCIDS identify, assess, and prioritize joint military capability needs (DoD, 2007).

The JCIDS implements a capabilities-based methodology that "leverages the expertise of all government agencies to identify improvements to existing capabilities and to develop new warfighting capabilities" (DoD, 2007). This approach requires a collaborative process that utilizes joint concepts and integrated architectures to identify prioritized capability gaps and integrated policy approaches to resolve those gaps (DoD, 2007). Once the required capabilities (what we want to be able to do) are identified, the JCIDS process is intended to assess our capacity to fulfill those capabilities.

Systems architecting and systems engineering present complementary approaches to the development of an SoS. For capability-based development of unprecedented systems, the initial portions of the traditional SE process have been demonstrated to show unsatisfactory results, in particular due to the complexity in transforming ill-defined capabilities into requirements useful enough to begin any engineering-based design. A capability is defined as "the ability to achieve a desired effect under specified standards and conditions through combinations of means and ways to perform a set of tasks" (IEEE, 2006). "Capabilities, often referred to as operational scenarios, consist of a sequence of operational activities needed to respond to or to provide an external stimulus" (Whitcomb, Vaidyanathan, et al, 2008). In the typical SE process, the capabilities-based development process incorporates a capability pull feedback loop. This will ensure customer capability needs are continuously being addressed and revised throughout the SE process.

The outcome is the fundamental description of the basis for the system—its architecture. This architecture defines the elements, their relationships, and the principles guiding its design and evolution. This architecture must be made visible to all system stakeholders, since it is the first embodiment of the system that can be reasoned about.

UNITED STATES NAVY SALVAGE COMMUNITY

Salvage forces have unique tasks, which require specialized equipment and highly trained personnel. The "triad" of U.S. Navy salvage forces integrates the Mobile Diving and Salvage Unit (MDSU), Military Sealift Command (MSC), and the Supervisor of Salvage and Diving (SUPALV, NAVSEA 00C), and serves as the core for removing hazards of navigation (in foreign and domestic coastal waters), repair and towing damaged vessels, recovery of sensitive items (such as aircraft black boxes), and recovery of other high-value objects from the ocean depths.

The salvage platforms of the Navy have become a vital part of a wide range of military operations, as highlighted in this chapter. The life cycle of the current towing and salvage ships are soon coming to an end, with a need for a future possible single-hull salvage platform within the next 10 years. At this time, an SoS architecture has not yet been developed to attain the proper systems requirements, based on the community's needs. The architecture elements data for the towing and salvage community has not yet been entered into the current NAERG metrics. All other system functions, activities, elements, and nodes that have not yet been recorded, have been entered into the towing and salvage architecture for the purpose of this paper.

United States Navy salvors depend on the systems needed to successfully complete their missions, maintain equipment functionality, and operate in extreme environments. To enable timely and reliable mission completion, individual systems must work efficiently together as an SoS. Desired requirements have been proposed by SEA 00C based on current ship capabilities, and require a capability-based requirements generation process, based on the current and future mission objectives.

MISSION ANALYSIS

The DoD must ensure that the towing and salvage community is developing systems to accomplish their assigned missions in a timely manner and positioning them accordingly. In order to provide the salvage platform baseline for trade studies to establish a CONOPS for salvage SoS design, the DRM concept is used. A well developed DRM will facilitate generation of requirements and subsequent system design" (Skolnick & Wilkins, 2000).

"For the government led development process, the DRM feeds the development and certification of a system functional baseline and provides support through the entire life of the program. Thus the DRM must support the program throughout the systems engineering process" (Skolnick & Wilkins, 2000). To ensure that the final iteration of the DRM is the best solution for capabilities-driven requirements generation, it is important to receive feedback from all actors associated with the system and then to refine the DRM based on that feedback.

In order to complete the mission success levels, all T-ARS(X) salvage, towing, heavy lift, diving, and pollution response capabilities or operational activities will be utilized. Each mission included within a DRM scenario can be decomposed into the individual operational activities necessary to complete the tasks that the DRM scenario requires. The DRM is decomposed into the following operational activities:

- Towing.
- Salvage (including Heavy Lift).
- Diving.
- Pollution Response.

Once all operational activities or capabilities have been identified, the components required to achieve the functions necessary to complete the mission will be identified and documented.

ARCHITECTING PROCESS

The purpose of this section is to describe the steps taken to develop an SoS architecture, from the mission design to the system specification, with the aid of an architecting tool. The development of an architecture using CORE is defined for the future salvage platform SoS and adequately identifies the capability-based requirements in terms of the operational mission objectives. This section also illustrates how a U.S. Navy Diving and Salvage system could be architected in the context of an SoS from an identified set of stakeholder needs.

The MBSE integrated methodology is used to select the SoS architecture. The identification of the salvage platform SoS begins with the mission objectives from the CONOPS of the salvage force, to developing a DRM, and leading to an appropriate architecture supported by modeling and simulation. The framework used in the development of the SoS architecture is modeled in CORE, and will be used as a foundation for future capability-based architectural modeling for the future salvage platform force structure (Huynh & Osmondson, 2007).

CORE 5 Architecture

A major challenge in the architecting process is developing an architecture so that the system elements are complete and consistent with one another. An architecting tool is a great asset that is used to verify that the data is consistent and that all element connections remain with their associated counterparts. The amount of data, when complying with architecture standards such as the DoDAF, is too large to manipulate manually.

CORE uses a unified model that integrates well with the architectural frameworks making the SoS architecture development process and the element relationship representation straightforward. “The CORE product suite is a fully integrated, flexible approach to a collaborative product design specifically developed by systems engineers for systems engineers” (Vitech, 2007). CORE delivers a mutual design-centric approach to product development. “CORE provides comprehensive traceability from need definition through requirements and analysis to architecture and test. Built upon a proven approach and a central integrated design repository, CORE includes a comprehensive behavior modeling notation” (Vitech, 2007).

Operational models are developed using MBSE principles. The design activities integrate the operational model and the systems model, and consist of requirements analysis, functional analysis, physical architecture synthesis, and verification and validation (Vitech, 2007).

The architecture is divided into two domains: operational architecture and system architecture. Each domain is described in detail below. “The Operational Architecture Domain captures originating concepts, capabilities, and supporting operational analysis to exploit, whereas the System Architecture Domain expresses the requirements, functions, and components comprising the physical design” (Vitech, 2007). The CORE architecting schema separates the systems and operational domains with relationship lines connecting the individual elements.

The primary CORE architectural elements used in this study are the Architecture, Operational Nodes, Operational Activities, Missions, Functions, Requirements and Components. From these elements, the necessary DoDAF architectural views and system specification document can be formulated. This chapter will describe in detail the individual elements, as well as how they relate.

Naval Architecture Elements Reference Guide

Architecture implementation is best organized around standard semantics and reference terminology (Ring, 2001). “Every architecture should be constructed from common terms, forming the elemental building blocks of the architecture, standardizing architectural elements” (Naval Architecture Elements Reference Guide, December 2007).

“Architecture elements represent the critical taxonomies, requiring concurrence and standardization for an integrated architecture as described by the DoDAF” (Siel, 2007). They contain the diction for the architectural views and are used to ensure a consistent integration of systems within an SoS architecture. “The data contained in the Navy Architecture Element Reference Guide (NAERG) shall be used for overall architecture framework development, programmatic research, development, and acquisition activities, and related integration and interoperability and capability assessments” (Siel, 2007). CORE has the advantage of having all of the NAERG lists available in the software.

The Supervisor of Salvage (SUPSALV) SoS enterprise will be described in terms of the NAERG elements, in order to explicitly define the architecture. The SUPSALV NAERG elements are organized into the following lists:

- Common System Function List (CSFL)
- Common Operational Activities List (COAL)
- Common Information Element List (CIEL)
- Common Operational Nodes List (CONL)
- Common Systems Nodes List (CSNL)
- Common Systems List (CSL)

As many COAL, CSFL, CONL, and CSL elements as possible were taken from the NAREG in CORE. Any not found in the NAREH were added, as needed.

Architectures

"Architectures exist for the purpose of achieving a well-defined system in both the operational and system domains, for a specific time frame. The Architecture class is used to identify an architecture and its time frame" (Vitech, 2007). Nodes in the systems architecture are defined as components, while nodes in the operational architecture are defined as operational nodes. For the towing and salvage platform model, the architecture was created as "Towing and Salvage".

Operational Architecture. Given the need to comply with the framework of the operational requirements document, the systems engineer must define the operational behavior in order to accomplish the mission. The operational architecture organizes the architectural elements, which compose the operational behavior of the system. The operational architecture is made up of the operational nodes, operational activities, operational tasks, and missions. Creating an operational architecture begins by first defining the mission, and then by identifying the operational activities needed to accomplish the mission. Once all of the operational activities have been identified, the responsible operational nodes can then be defined.

Operational Nodes. "Within the Operational architecture domain, the operational node is part of the operational context which also includes the elements that represent the external aspects of the operational domain" (Vitech, 2007). An operational node is a representation of an actor role within an organization that produces or consumes information. The operational nodes for the future towing and salvage platform are all of the actors/organizations that interact with and make decisions for the system. They include:

- SEA 00C (SUPSALV) including all departments.
- MDSUs One and Two.
- ESSM.
- MSFSC.

The operational nodes can be decomposed and displayed in CORE as a system diagram. Further breakdown of the operational nodes would characterize operational activities.

Operational Activities. Operational activities, also called operational scenarios, consist of a sequence of capabilities needed to respond to an external stimulus. Each operational activity is performed by an element within the operational node class. Finalized capabilities (operational activities), are incorporated to become the integrated model for the architecture.

The operational activities are linked to the systems architecture domain through the function element, and are traced from operational nodes and achieve operational tasks and missions, as displayed in Figure 2. "Operational activity traceability from an appropriate mission element is established using the 'achieves' relationship. Establishing this relationship enables one to easily assess what capabilities are impacted by a mission change and what missions are impacted by a capability change or failure" (Vitech, 2007).

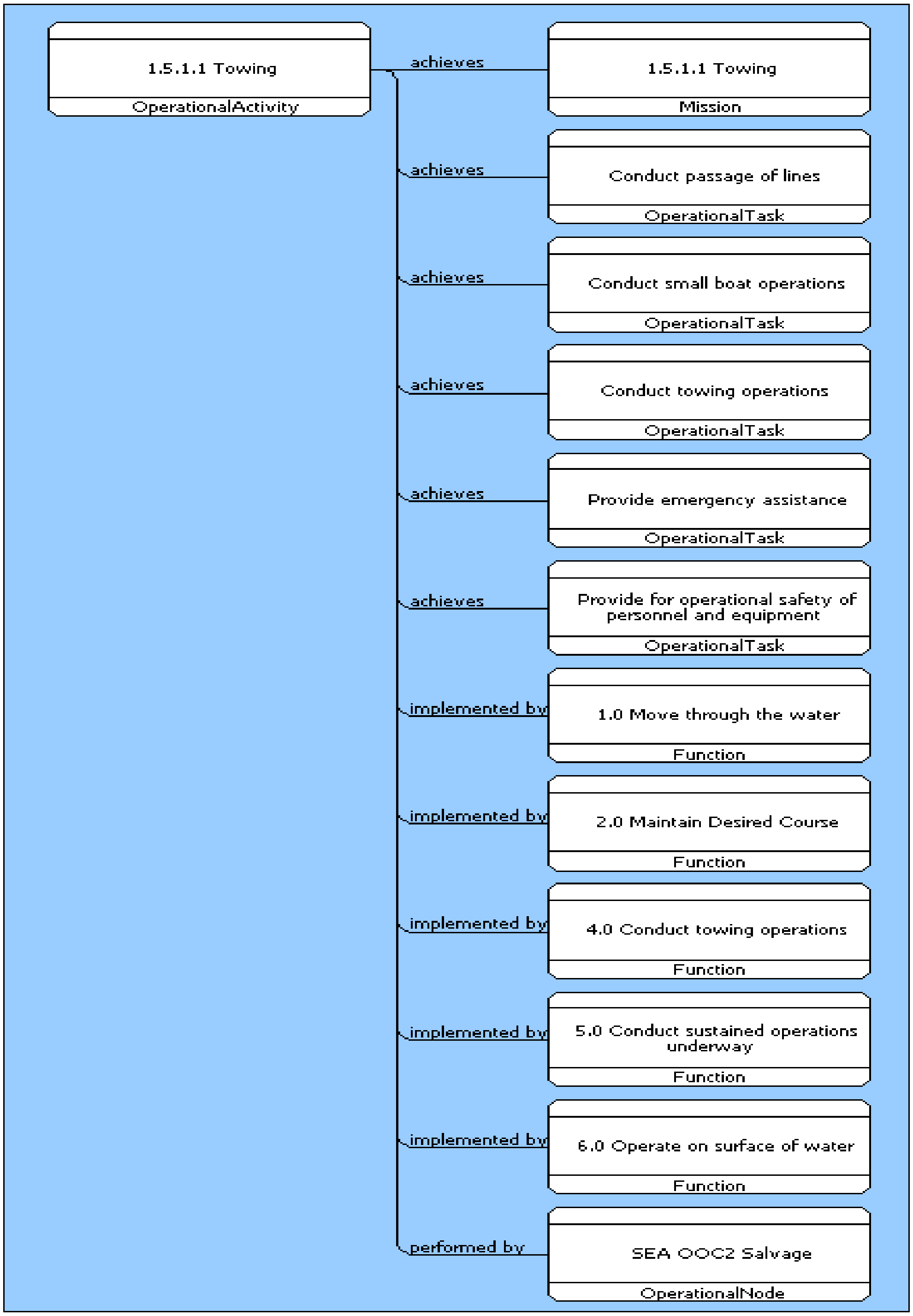

Figure 2. CORE systems view of the operational activity "Towing"

Required Operational Capabilities (ROC). Required Operational Capabilities (ROC), as constituted by mission commanders, detail the capabilities required of ships in various operational situations outlined in the projected operating environment (POE). The level of detail is decomposed to outline specific mission areas and component/operator responsibilities. The ROC provides the necessary details of operational capabilities for which the ship class was designed, based on expected missions. It will establish tasking that produces a measurable workload used to compute manpower requirements.

The ROC is further decomposed into operational tasks needed to fulfill the operational activity. For example, the operational activity "Mobility" is composed of lower-level activities such as "move through the water" and "conduct sustained operations underway." Each of these activities can be further decomposed into individual tasks necessary to achieve the activity "move through the water."

Operational Task. The operational task element decomposes a list of mission-derived tasks with associated conditions and standards that a system architect may select to accomplish a simulated mission. The Universal Naval Task List (UNTL) is a combination of the Navy Tactical Task List (NTTL) and the Marine Corps Task List (MCTL), and was utilized to identify the universal tasks that the towing and salvage platform must perform.

Along with the UNTL, there are task lists derived from a hierarchy of DoD tasks contained within the Universal Joint Task List UJTL. Depending on the mission level being developed, a certain standard of tasks are required to fulfill that mission-level requirement. If the mission involves joint service cooperation, the tasks would be derived from the UJTL at a higher-level mission perspective.

In order to complete the mission requirements, the type of operation must be considered. Each mission will require a unique set of capabilities or operational activities due to the variation of the mission environment. Task lists are uniquely defined based on a higher-lever mission analysis of the variation in operational objectives. Although many of the tasks within the different lists are similar, task requirements will vary based on the type of operation.

As stated above, the task list identifies "what" is to be performed in terms of the system being designed. The following towing and salvage tasks were derived from the UNTL for the purpose of developing a CORE architectural model, which included: provide damage control, conduct small boat operations, sail ship from port, anchorage, or moorage, return ship from port, anchorage, or moorage, employ remote vehicles, conduct navigation, conduct ship-to-shore or ship-to-objective maneuver, conduct sustained operations ashore, conduct security, conduct passage of lines, transport personnel, transport cargo, provide support services, employ communication security, coordinate damage control operations, conduct personnel recovery, perform search and rescue, provide disaster relief, provide emergency assistance, provide for operational safety of personnel and equipment, conduct towing operations, conduct salvage operations, retract beached vessels, conduct off-ship firefighting, conduct heavy lift operations, conduct diving operations, conduct mooring, conduct underway replenishment.

These tasks were derived to satisfy the capabilities needed in order to perform the higher-level tasks included in a simulated ROC/POE developed within the DRM. These tasks were used to identify the required operational activities necessary to complete the proposed DRM and further recognize the operational nodes responsible to meet mission needs.

Missions. The problem(s) must be defined well to ensure the development and refinement of the correct data necessary to address the situation. The problem definition step is accomplished by developing a system architecture that achieves a reference mission, to which the operational activities of the system will need to be demonstrated within a mission simulation. In CORE, the element relationship for the decomposition of the mission element was derived and is displayed in Figure 3.

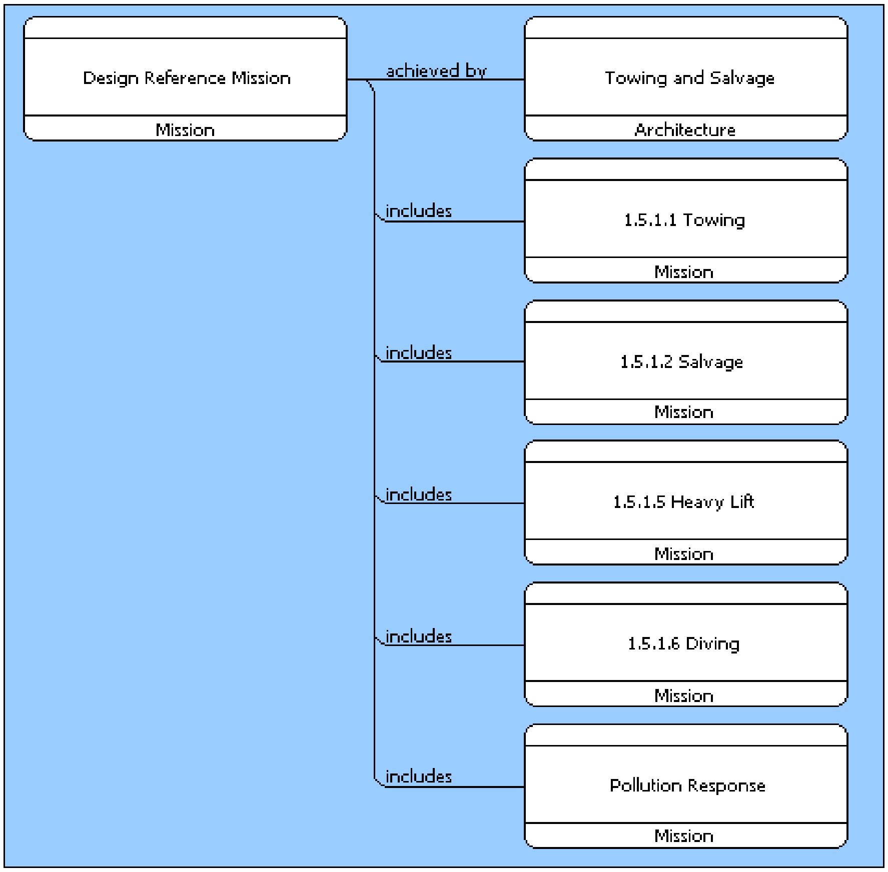

Figure 3. CORE view of the DRM decomposition

Systems Architecture Considerations. SE activities needed to complete the architecture and interrelate the operational and systems domains are developed through the systems performance parameters, with the integration of the component and function elements as a basis of the requirements (Vitech, 2007). The components with respective functions are derived from the operational activities needed to perform the mission. The example component type service is built from a system component to perform a service function.

Functions. Functional decomposition refers to the process of organizing the functional relationships into its components or systems for the purpose of defining the identity of the components. Specifically, what function must be provided to accomplish the mission requirements and how will that function be fulfilled by use of a system component?

A function is the property of a system that, when performed, will fulfill a requirement for an objective. Functions are decomposed into lower-level functions (see Figure 4), until the individual components can be traced to a particular function to be performed. Functions are based on requirements that can be identified in the beginning stages of system development as desired characteristics. The functions identified for the towing and salvage platform are based on all the operational activities required to achieve the missions that the towing and salvage community is required to perform.

Components. An objective of the system architecture is to identify what are its critical components and what are the relationships between all components within the system. The components identified in this architecture range from higher-level systems like “ship” to lower-level individual components like “Diver davits.” Each component is organized within the Ship

Work Breakdown Structure (SWBS). All towing- and salvage-related components were entered into CORE, and organized into their respective SWBS groupings.

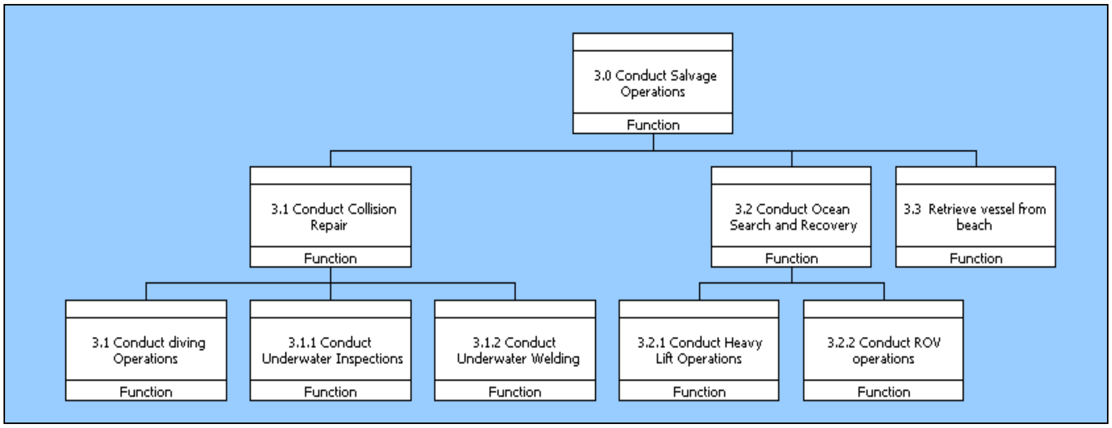

Figure 4. CORE view of the functional decomposition of "Conduct Salvage Operations"

Functional Requirements. Requirements are the basis of a function and usually specify the goals of the system. "Requirements development occurs when operational activities and performance characteristics serve as sources for system requirements" (Vitech, 2007). Operational activities lead to the identification and definition of functional requirements that, when added to the identification of performance characteristics, results in system requirements. Thus, a requirement is a result of an operational activity and a performance characteristic.

The requirements generated from a capability-need, MBSE methodology are a complete set of requirements that will be a basis for the system specification document. The generated requirements were compared with the given set of requirements from SUPSALV, to produce a comparative analysis of requirements-based system modeling versus capability-based system modeling.

The primary/critical performance requirements given by SUPSALV match the high-level, capability-based requirements that were generated by the CORE tool. The CORE tool can produce a complete capability-based requirements list, with all mission-based functions accounted for and mapped to all respective lower-level components.

Nonfunctional Requirements. Nonfunctional requirements identify criteria that can be used to evaluate the system's operational behavior instead of identifying specific functions of the system. In general, nonfunctional requirements define how a system is supposed to operate rather than what it is supposed to do. Nonfunctional requirements are sometimes referred to as "ilities," e.g., availability and survivability, which describe the criteria in which the system can be evaluated. Within the CORE architecting tool, nonfunctional requirements are not present within the schema, but are present within requirement class with the type attribute set to "Constraint."

The process starts with extracting the originating requirements into the requirements class and then set the "type" attribute to (Functional, Performance, Constraint, or Verification). A Functional requirement will be modeled with "Function" and the nonfunctional requirements (except for performance) will be addressed by one of the specialty engineering disciplines.

Nonfunctional requirements will be clearly defined and utilized when creating a simulation based on the CORE model. The availability and/or survivability of a system cannot be determined without being able to simulate all of the components working together within an SoS, to include the environment.

Towing and Salvage Specific Methodology

The CORE architecture schema has many other elements which connect with and influence the interoperability of the architecture. The focus has been on the major elements which directly influence the capabilities of the system based on the previously defined mission. The major elements, once defined, can be used to generate the necessary architectural views that will allow stakeholders to communicate in their own terminology related to the future towing and salvage platform architecture. These elements, along with the architecture process steps taken for a comprehensive architectural view development, are displayed in Figure 5.

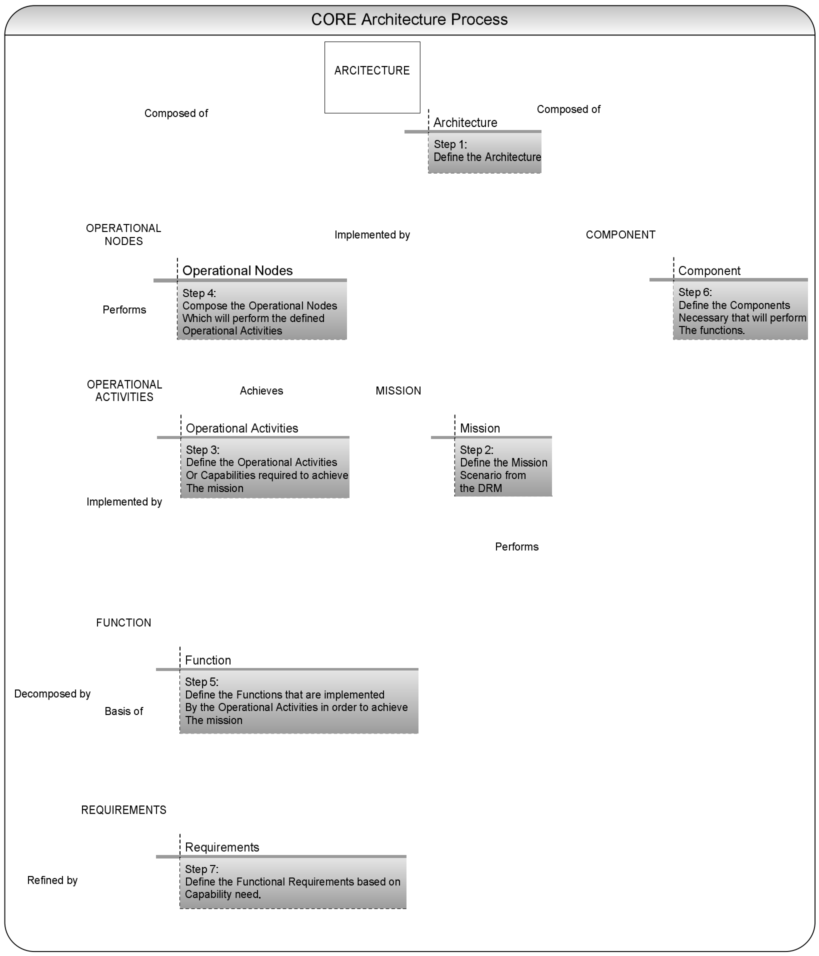

Figure 5. Methodology for CORE Towing and Salvage architecture development

The steps taken in the architecture process displayed in Figure 5 are based on a methodology built from the DRM capability need. The mission requirements are generated by a DRM that would require the combined capabilities of both towing and salvage platforms. Beginning with defining the architecture, the DRM was developed to incorporate the full functional potential of the towing and salvage activities. The next step in the process was to define the operational activities necessary to achieve the identified mission requirements, as well as link them to the operational nodes responsible for conducting those activities. The activities are also built from, and decomposed by, the standardized operational tasks linked to the individual mission tasks. Once the activities are identified, a functional requirements generation process is initiated, based on a functional hierarchy from the components necessary to complete the mission tasks. Finally, all elements are then redefined, decomposed, and linked to their schema element relationships.

ARCHITECTURE RESULTS

The DoDAF format displays and organizes a complex systems architecture into consistent views, showing interoperability between the system elements. Representations for the DoDAF products are created within CORE from the Entity-Relationship Diagrams (ERDs).

CORE documents the architecture products in a Rich Text Format (RTF), via scripts that generate any standard DoDAF product. The DoDAF version 1.5 product scripts are designed to be flexible in order to support any later iteration (Vitech, 2007). Figure 6 displays the integration of the SE process steps with each DoDAF view production ability based on time.

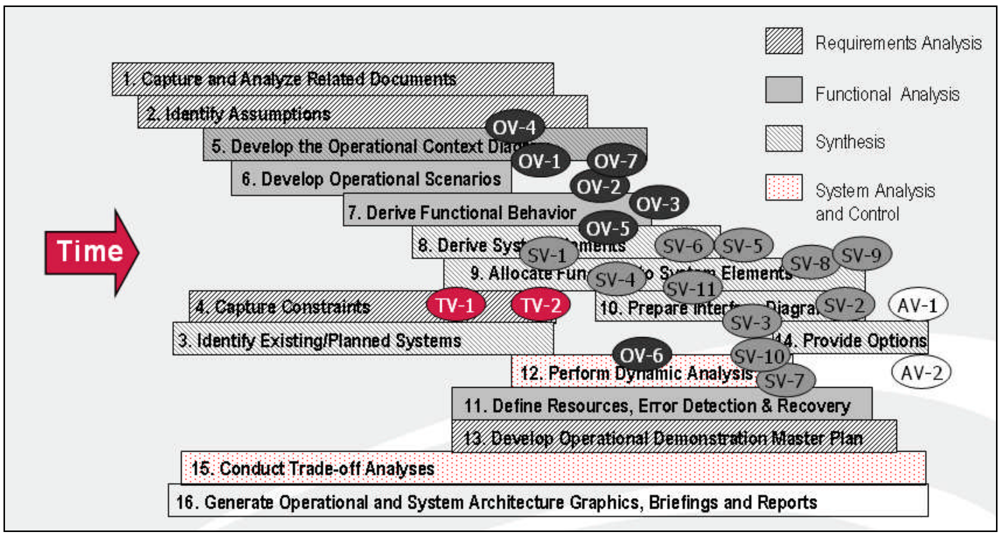

Figure 6. CORE integration of the typical systems engineering process with DoDAF views milestones (Vitech, 2007)

From an early stage SE development perspective, the DoDAF architecture views OV-2, OV-5, and SV-4 are the most important views because they lay the foundation for the operational architecture (structure, behavior, interfaces) and provide a basis for communicating the system architecture. For the purpose of this paper, only the OV-2, OV-5, and SV-4 are developed and discussed. A System Design Document (SDD) was also created during this study.

The SDD describes how the functional and nonfunctional requirements and CONOPS are transformed into system design specifications. The SDD developed with this architecture is a high-level, first-iteration document that records the system design through detailed design specifications. The SDD gives a high-level overview of the system architecture and is a formal documentation process for requirements generation that can be used to design the new towing and salvage platform. More component-specific results can be obtained from the SDD, which was also used as the detailed design reference for requirements generation.

Operational views detail the user's operating domain in which the developing system will operate (Zachman, 2007). The OV-2 is an operational node connectivity description, which displays the relationships between the nodes as well as organizes the nodes into an operational hierarchy. The operational node relationship hierarchy from SEA 00C is displayed in Figure 7.

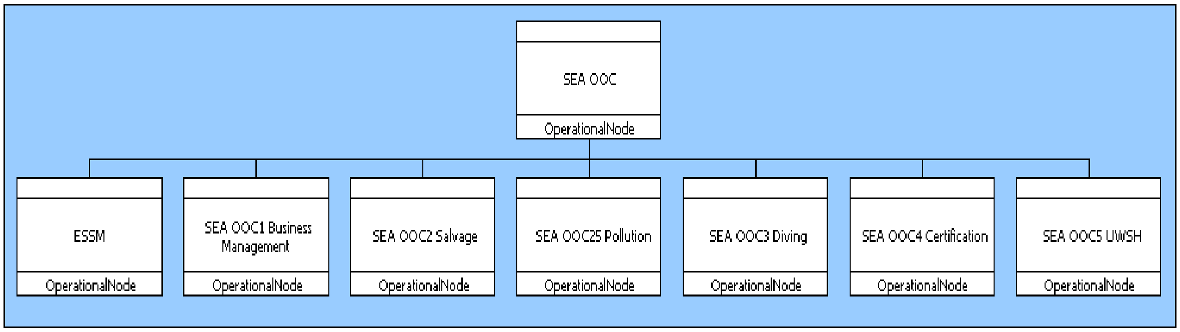

Figure 7. SEA 00C SV-2 architecture view

The OV-5 DoDAF view is an activity model that identifies and displays the hierarchical decomposition of an operational activity, as well as show the relationships between the capabilities and activities in which each activity is interconnected. The OV-5 Activity Model for Conduct Towing and Salvage DRM hierarchy is displayed in Figure 8.

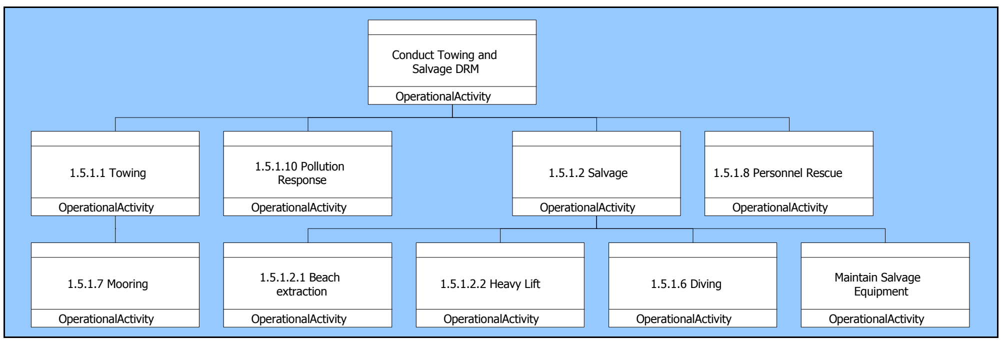

Figure 8. "Conduct Towing and Salvage DRM" Hierarchy Diagram

The Conduct Towing and Salvage DRM IDEF0 diagram illustrates the children or offspring operational activities with the user-selected operational nodes. This operational activity model graphically organizes the activities in a hierarchy, clarifying the level at which each function is required. Figure 9 is the IDEF0 diagram, depicting which operational nodes perform the "Conduct Towing and Salvage DRM" operational activities. The overlap within the activities demonstrates the actions performed by the operational nodes.

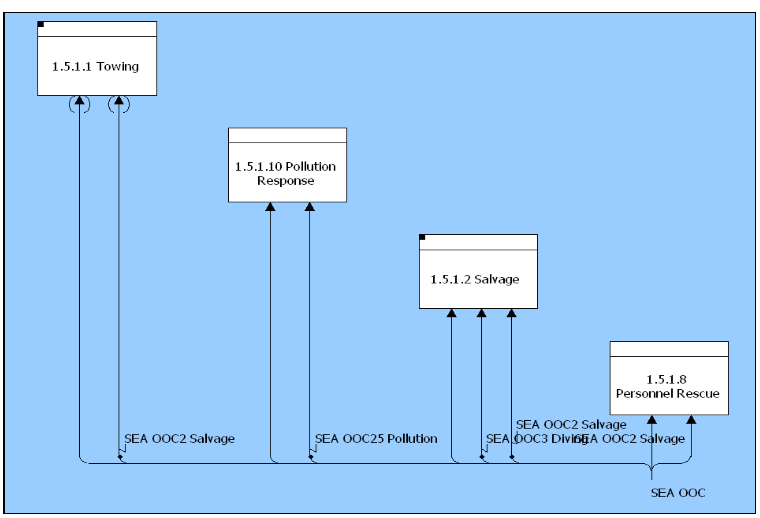

Figure 9. "Conduct Towing and Salvage DRM" IDEF0 Diagram

Within the OV-5, each of the children operational activities can be analyzed, along with their activity relationships among their corresponding operational nodes. Figure 10 displays the IDEF0 diagram of the "Salvage" operational activity.

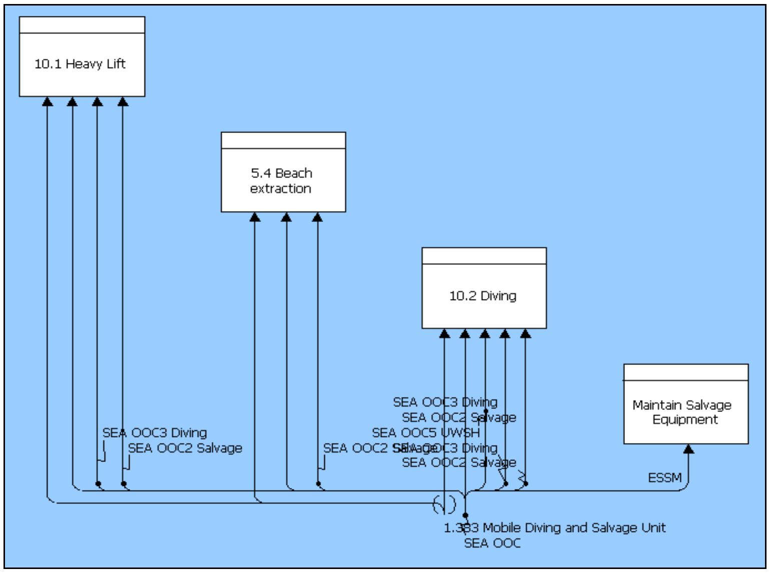

Figure 10. "Salvage" IDEF0 Diagram

The DoDAF system and service views are a set of graphical products that describe systems and interconnections that support DoD functions. SV products focus on specific systems with specific physical locations. "The relationship between architecture data elements across the SV to the OV can be exemplified as systems are procured and fielded to support organizations and their operations" (DoDAF, 2004). The system and service view focused on in this paper is the SV-4 view, which documents the system data flows between functions. Figure 11 displays the SV-4 hierarchy for the function "Conduct Towing Operations." When developing a complete architecture, the level of detail from a functional decomposition within the SV views will ensure sufficient system design.

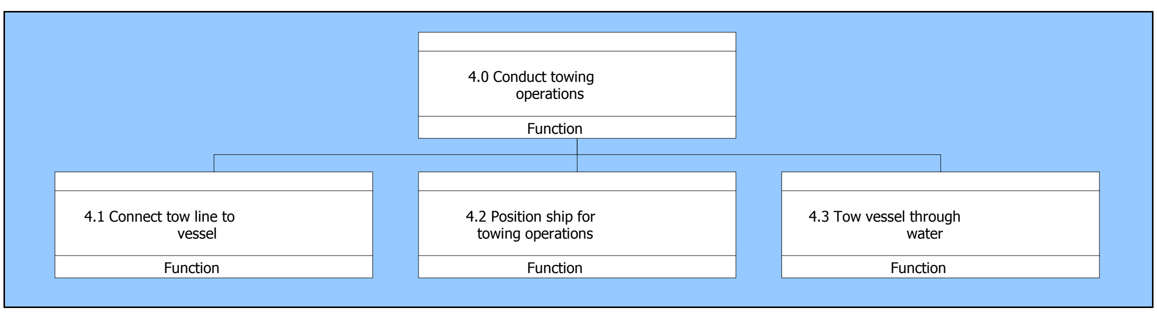

Figure 11. "Conduct Towing Operations" Hierarchy Diagram

This SV-4 documents the functional relationships of just one of the functions within the system and can be expanded to include all system functions. The "Conduct Towing Operations" function can be displayed with the component relationships necessary to achieve the functional requirement. The IDEF0 context diagram is displayed in Figure 12.

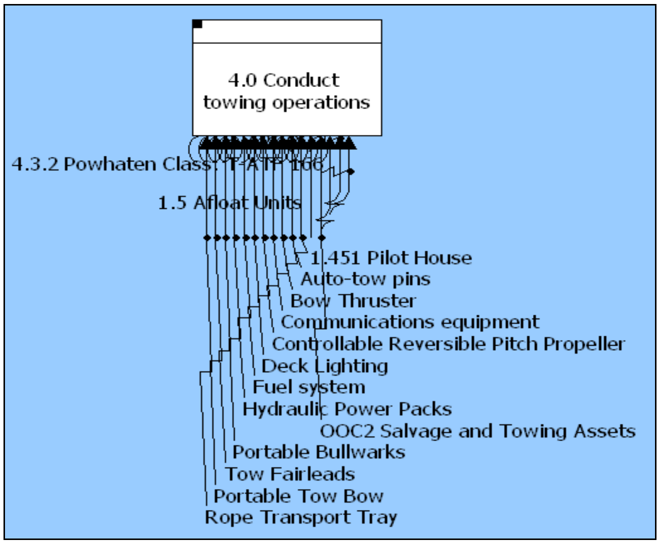

Figure 12. “Conduct Towing Operations” IDEF0 A-0 Context Diagram

The SV-4 function can be decomposed, showing the functional breakdown of the “Conduct Towing Operations” function with the component-to-function individual relationships. Figure 13 displays the IDEF0 diagram of the functions necessary to perform the “Conduct Towing Operations” function.

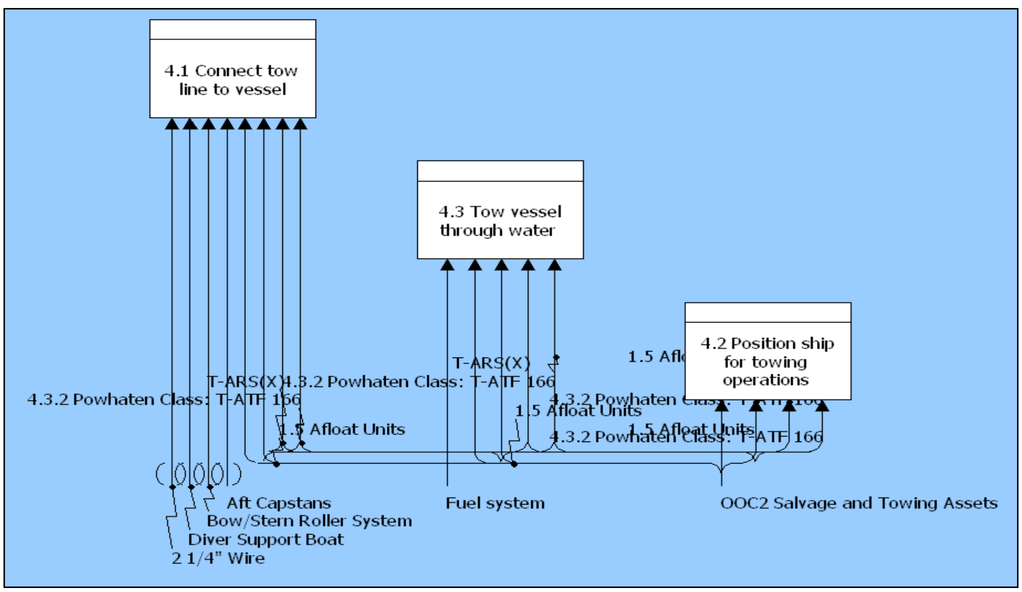

Figure 13. “Conduct Towing Operations” IDEF0 Diagram

From the fundamental architecture and DoDAF products generated, a ship design team can begin a concept design phase based on the requirements generated with function-to-component relationships defined. Lower-level requirements generation must be developed in order to generate a complete analysis of the buy versus build options.

Commercial Towing and Salvage Market Analysis

A top-level commercial market analysis, mapping the capabilities to the architecture is described below. In CORE, these commercial capabilities are captured as resources for a potential T-ARS(X) simulation. Figure 14 displays the top-level, T-ARS(X) requirements compared to the current commercial capabilities. Not all desired or derived requirements were analyzed due to either a lack of available information or the naval architecture design synthesis needed to obtain a quantities value.

Vessels of Opportunity	Speed 15/20	Bollard Pull 125/175	Navy Personnel Accomodation 42; T=0	Civilian Crew Accomodation 15; T=0	Positioning DP-2/DP-3	Endurance 8,000 nm @ 8 kt / 12,000 @ 10 kt	Unobstructed Deck Space; Aft 3600 sqft / 4300 sqft	Crane; Lift Capacity Min. 120 short tons SWL Amidships / Amidships + Stern	Crane; Fwd 10 short tons SWL; T=0	Ice Classification ABS Ice Class C0; T=0
1 Noordhoek Diving Construction support vessel	12 knots	44mT	55	15	DP-2		7373 sqft	110.2 sh	30 mT	
2 Trans Viking AHTS/VS 4622 CD	12 knots	220 ts	185	15	DP-2		8223 sqft			
3 Rolls-Royce UT-design UT 719-2		85 ts	185	15	DP-2		4477 sqft	5 ts		
4 Gulf M.V CLWYD Supporter	14 knots	105 t					3907 sqft	6 ts		Ice-C
5 Merwede Seven Atlantic Diving Support Vessel	13.6 knots		135	15	DP-3		12916 sqft	150 mT	10 mT	
6 Merwede HOS Superior Acheiver	13 knots		84	15	DP-3		16145 sqft	140mT	160 mT	
7 Nomis DEA Olympian	13 knots	150 t	30	12			4305 sqft			
8 Nomis DEA Signal	12 knots	150 t	12	12			4660 sqft	2.5 t		
9 Noordhoek Singapore	12 knots	38 t	21	15	DP21		3336 sqft	25 t		
10 Sealion Tiosa Pegasus	13 knots		84	15	DP21		12916 sqft	400 mt	7 mt	
11 Sealion Toisa Paladin	15.5 knots		85	15	DP-2		9687 sqft	140 t	30 t	
12 Sealion Toisa Polaris	13.25 knots		83	15	DP-3		7534 sqft	150 t	10 t	
13 Sealion Toisa Proteus	15 knots		145	15	DP-3		20451 sqft	196 t	19 t	
14 Ulstein A102	17.5 knots	200 t	20	15			5100 sqft			
15 Ulstein P101	16 knots		45	15			11500 sqft	100 t	0	
16 Farstad BOS Topazio	15.5 knots	150 t					6000 sqft	10 t	10 t	
17 Farstad Far Saga	15.5 knots						6200 sqft	100 t	20 t	
18 Farstad Far Sapphire	17 knots	261 mt					8500 sqft	20 t	3 t	

Figure 14. Commercial capabilities with requirements comparison

The vessels of opportunity are derived from a complete list of available commercial platforms and are considered to be the closest match to the generated requirements. All cells highlighted in blue either meet the requirement or surpass the lower limit of the requirement. Yellow is close and can be improved, while red will not meet the requirement, even with improvements. Bollard pull and crane lift capacity are two contracting requirements that seem impossible to met simultaneously. Not all of these platforms are designed to tow, but have the ability if configured correctly. The platforms that do not have ample crane lift ability can be configured with an additional crane to meet that requirement. If any of the requirements cannot be met, based on T-ARS(X) mission need as mapped in the architecture, the joint towing and salvage capability cannot be achieved. If the missing information can be determined, then several platforms may be candidates.

All of the published performance requirements/characteristics were based on a perceived mission need and were documented as an estimate for future use. CORE can provide accurate requirements traceability documentation based on mission need. Components are the lower-level elements of the systems architecture. The functions are performed by components and are necessary to achieve the mission capability. The SUPSALV desired characteristics are verified to fulfill a mission need in CORE. Table 1 displays how each desired T-ARS(X) characteristic should be determined.

	Desired Performance Characteristics	
Characteristic	(Threshhold/Objective)	Should be obtained from a MBSE architecture
Speed	15/20	Based on sea basing/DRM
Bollard Pull	125/175	Based on function implements capability
Navy Personnel Accomodation	42; T=0	Based on component architecture simulation
Civilian Crew Accomodation	15; T=0	Based on component architecture simulation
Positioning	DP-2/DP-3	Standard ship system requirement
Endurance	8,000 nm @ 8 kt / 12,000 @ 10 kt	Based on sea basing/DRM
Unobstructed Deck Space; Aft	3600 sqft / 4300 sqft	Based on mission needs
Crane; Lift Capacity	120 short tons SWL	Based on function implements capability
Crane; Fwd	10 short tons SWL; T=0	Based on function implements capability
Ice Classification	ABS Ice Class C0; T=0	Based on sea basing/DRM
Stability	Adequate metacentric height, 30 yr service life	Standard ship system requirement
Unobstructed Deck Space; Fwd	720 sqft; T=0	Based on mission needs
Survivability	Commercial Salvage Standards, ABS Classification	Standard ship system requirement

Table 1. Desired performance requirements mapping process

Based on the analysis of the commercial market compared to the MBSE-generated requirements, a top-level system design with consideration of available capabilities has been developed. The architecting process, with implied T-ARS(X) characteristics, has been mapped to a verifiable set of requirements for future ship design. The results indicate a gap in bollard pull and crane lift capacity for commercial platforms. The final step in this analysis would be to analyze the cost comparison of outfitting the missing requirements on the commercial platforms with building a new platform.

Summary

The architecture demonstrated highlights some of the more important DoDAF products, but they barely scratch the surface of the potential towing and salvage architecture development available in the CORE modeling tool, for a final SoS development. The SDD is one of many documents that can be produced by the push of a button, once the elements have been completely defined and linked accordingly. Capabilities-based architecting approach for the recapitalization of the future towing and salvage platform has been demonstrated, providing a high-level/first-iteration of requirements generation.

Future T-ARS(X) operations will require an unprecedented level of integration among joint towing and salvage capabilities. The towing and salvage community's increased demand for a mission-tailored future salvage platform requires a more integrated approach to T-ARS(X) requirements generation. Along with a towing and salvage force simulation, MBSE can achieve a comprehensive platform design for either build or buy recapitalization strategy.

This paper has defined a process for realizing desired capabilities based on mission requirements. The need for MBSE has also been presented and a model-based architecting process has been developed, based on the fusion of systems architecting and SE. The differences involved in systems architecting have been presented, showing the benefits that it can bring to a system design.

LIST OF REFERENCES

Defense Acquisition University (DAU) (2001, January). "Systems Engineering Fundamentals."

Department of Defense. (2007, 1 May). "Joint capabilities integration and development system." CJCSI 3170.01F

Department of Defense. (2005). "Joint capabilities integration and development system." CJCSI 3170.01E, 2005.

Department of Defense, (DODAF) (2004, February 9). "Architecture framework, Vol. I: Definitions and guidelines."

Fein, G. (2008, August 21). "Navy developing path forward for open architecture implementation." Defense Daily

Huynh, T.V., & Osmundson, J.S. (2007). "An integrated systems engineering Methodology for analyzing systems of systems architectures."

IEEE 1220 (2006)

Maier, M & Rechtin, E (2002). "The Art of Systems Architecting." *CRC* Press.

Skolnick, F.R., & Wilkins, P.G. (2000). "Laying the foundation for successful systems engineering."

Sperling, R., & Keenan, J.D. (2006, November). "Navy salvage ship recapitalization study," CRM D0014544.A1/Final.

Vitech Corporation. (2007). "CORE 5 Architecture Definition Guide."

Whitcomb, C., Vaidyanathan, R., Pluta, J., Kiihne, J., & Solitario, W. (2008, 6-8 April). "Architecture development for an unmanned vehicle sentry systems of systems." 23rd Bristol International Unmanned Air Vehicle Systems (UAVS) Conference, Bristol, UK.

Walker, S.K. (2005). "Capabilities-based planning – How it is intended to work and challenges to its successful implementation."