NPS-SE-21-233

ACQUISITION RESEARCH PROGRAM
SPONSORED REPORT SERIES

Development and Demonstration of a Capabilities Focused Model Based Systems Engineering Framework

December 1, 2020

Dr. Paul T. Beery, Assistant Professor

Graduate School of Engineering and Applied Sciences

Naval Postgraduate School

Prepared for the Naval Postgraduate School, Monterey, CA 93943.

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 the Acquisition Research Program (ARP) via email, arp@nps.edu or at 831-656-3793.

Abstract

This research develops a Capabilities Focused Model-Based Systems Engineering (CF-MBSE) Framework that demonstrates the relationship between MBSE architectural representations and operational modeling approaches. The role of MBSE products and operational models are discussed with respect to the Systems Modeling Language (SysML) and the Department of Defense Architecture Framework (DoDAF). This research develops a tailored systems engineering process, leveraging recent trends in MBSE, that results in early identification of desirable system capabilities. This process is flexible for systems of varied size, resulting in applicability to broad, fleet level issues, as well as investigation of design considerations for new DoD platforms.

Department of Defense acquisition relies heavily on systems engineering, with recent efforts emphasizing the role of MBSE to support system definition, analysis, and development. Recent work in MBSE has focused largely on development of SysML, in particular its role in executable systems architectures. While this formalization has proved valuable, an unintended consequence has been a disconnect between recent investments in MBSE capability and the systems engineering processes utilized in support of DoD acquisition programs, to include the DoDAF. This research presents a CF-MBSE Framework that formalizes an approach for utilization of MBSE in support of system operational capability assessment within system acquisition. This framework integrates multiple MBSE domains (to include Requirements Definition, Architecture Development, and System Modeling) and highlights the potential impact that MBSE can have in support of DoD system acquisition. This serves two purposes. First, for the DoD, the gaps between the current efforts in MBSE and DoD acquisition can be reviewed and assessed. Second, for the broader community, a process is defined and demonstrated that integrates the products currently produced in a typical MBSE effort into a formal, capabilities focused approach. To highlight the applicability of the approach, the CF-MBSE Framework is applied to an analysis of the next generation NATO main battle tank. That

demonstration results in the following: identification of initial system requirements, development of a comprehensive system architecture that describes both the anticipated operational employment and system functionality, development of an operational simulation that facilitates analysis of the system in a representative environment, and analysis of simulation results that identifies key system design characteristics and environmental conditions. Because the operational model and associated analysis are linked directly back to the previously developed system architectures (that present both an operational and system design perspective) those results are explicitly linked to SysML/DoDAF products and system design characteristics.

NPS-SE-21-233

ACQUISITION RESEARCH PROGRAM
SPONSORED REPORT SERIES

Development and Demonstration of a Capabilities Focused Model Based Systems Engineering Framework

December 1, 2020

Dr. Paul T. Beery, Assistant Professor

Graduate School of Engineering and Applied Sciences

Naval Postgraduate School

THIS PAGE LEFT INTENTIONALLY BLANK

Table of Contents

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

Introduction

Motivation

In January 2016, the chief of naval operations (CNO), ADM John Richardson, published *A Design for Maintaining Maritime Security*, guidance that framed the future direction of the U.S. Navy (Richardson, 2016). That guidance has been revised and expanded in *A Design for Maintaining Maritime Superiority 2.0* (Richardson, 2018). Both documents present guidance that informs the need to modernize engineering efforts. Specifically, *A Design for Maintaining Maritime Security* presents a set of objectives and lines of effort that have shaped recent efforts in systems engineering (Richardson, 2016). That guidance emphasized that there is a need and priority to "better meet today's force demands, explore alternative fleet designs, including kinetic and non-kinetic payloads and both manned and unmanned systems … (to) include exploring new naval platforms and formations" (Richardson, 2016, pp. 6). Currently, U.S. Navy systems commands are investigating the utility of systems engineering, in particular model-based systems engineering (MBSE), to support that design, analysis, and acquisition of systems in accordance with this instruction.

Generally, MBSE is being developed as an approach to the realization of systems when presented with complex problems. Specifically, the systems of interest to an MBSE project are often comprised of independent subsystems which, when coupled with the ambiguity of the requirements and system boundaries that typify engineering and acquisition efforts in the early stages of system design, often creates a demand, real or perceived, for a system or solution that is extraordinarily difficult to define, coordinate, and test. However, there has been substantial work concerning the development of MBSE tools that allow those challenges to be overcome.

The Systems Engineering Stakeholders Group (SESG), comprised of lead engineers from each of the naval systems commands, is working to identify and implement MBSE best practices, institutionalize MBSE in engineering technical reviews, and align training tools and standards for MBSE across the Navy. In support of those broader efforts, this research proposes development of a tailored systems engineering approach, leveraging recent trends in MBSE, that focuses on early identification of desirable system capabilities. This approach is motivated by and tailored to the need to explore alternative system designs and new platforms early in the system development life cycle, a tasking that is facilitated by a formal, Capabilities Focused-MBSE (CF-MBSE) approach.

Goals and Objectives

This research is expected to have immediate relevance via development of an engineering approach that utilizes recent trends in MBSE to focus engineering efforts in such a way that they can support designs that are specifically focused on operational capability. This has the potential to positively impact both the DoD acquisition and engineering processes. This research produces a formal, demonstrated framework that identifies potential connections between systems engineering and operational assessment approaches. In support of that broader goal, this project accomplishes the following objectives:

1. Development of a Capabilities Focused MBSE (CF-MBSE) Framework to support identification of desirable system configurations early in the design cycle. This will necessitate:
    a. Review of recent efforts in systems engineering and MBSE relevant to the fundamental concepts described in the CF-MBSE framework
    b. Discussion of operational capability assessment approaches appropriate for application to the CF-MBSE framework
    c. Integration of recent systems engineering and MBSE developments with operational capability assessment approaches
2. Demonstration of the CF-MBSE Framework through an analysis of a potential DoD relevant system.

Report Organization

This report presents an MBSE approach that is tailored for implementation to a broad set of potential systems. To highlight that intended broad applicability this report is organized into three primary sections. First, a literature review covering recent efforts in MBSE and operational modeling is presented. Second, a CF-MBSE framework that integrates each of those areas, including a summary of the fundamental concepts and steps associated with each phase of the framework, is proposed. Third, the CF-MBSE is demonstrated through analysis of a potentially applicable system. This research chooses a theoretical NATO main battle tank for demonstration. The results of each of those sections is summarized, and potential areas for future work are identified.

THIS PAGE INTENTIONALLY LEFT BLANK

Literature Review

Background

In June 2018 the Office of the Deputy Assistant Secretary of Defense for Systems Engineering published the *Department of Defense Digital Engineering Strategy* (Department of Defense, 2018b). That document, expanding the concepts presented in Zimmerman (2017) and Zimmerman and Dahmann (2018) is specifically tailored to support the goals and objectives identified in the National Defense Strategy (Department of Defense, 2018b). It describes the importance and relevance of modernizing engineering efforts to improve their applicability and support for DoD acquisition. Notably, the Department of Defense (2018b) states, "Current acquisition processes and engineering methods hinder meeting the demands of exponential technology growth, complexity, and access to information" (pp. 1).

This emphasis on revising engineering and acquisition processes to modernize acquisition is not unique to the Digital Engineering Strategy. Over the past 20 years, the DoD has invested in efforts such as simulation-based acquisition, synthetic environments to support acquisition, and digital twins. Uniformly, these approaches have emphasized that the use of a consistent data model is essential to developing a complete description of any system when trying to balance operational, design, scheduling, and cost perspectives. Many of the ideas developed throughout those efforts permeate engineering and acquisition strategy today. Frey and Valencia (2010) provide an overview of the role of modeling and simulation (M&S) as a strategy to support the systems engineering life cycle. Figure 1 provides a graphical representation of the support that M&S provides to systems engineering.

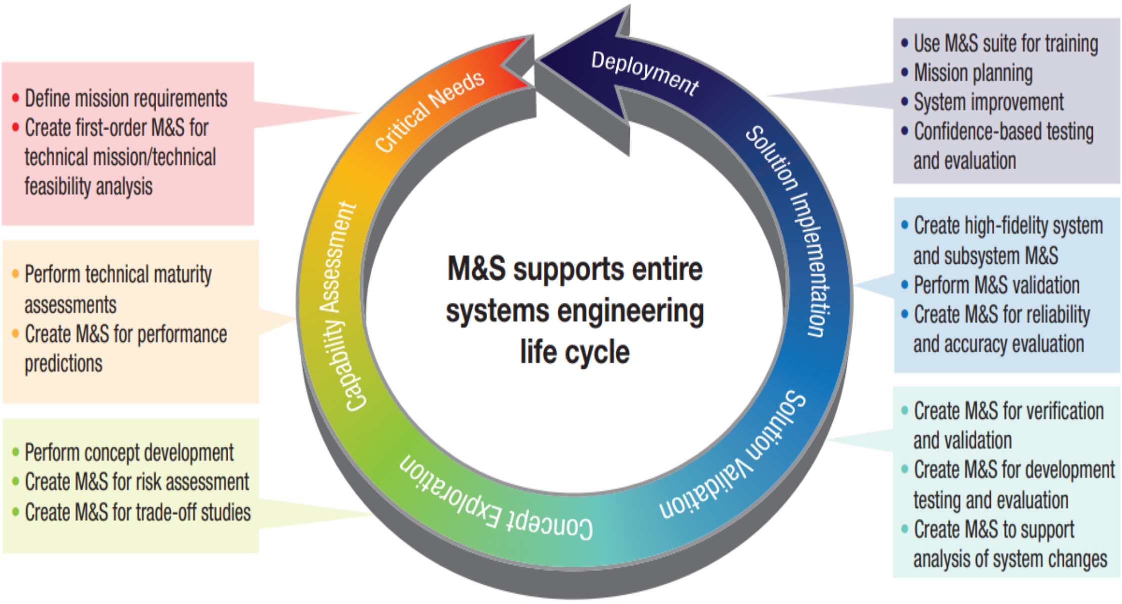


Figure 1. M&S Support for Systems Engineering (Frey & Valencia, 2010)

Note the broad potential applicability of M&S to support systems engineering efforts. Specifically, the left side of Figure 1 describes M&S as a tool to support definition of mission requirements, feasibility analysis, performance prediction, and trade-off studies. The right side of Figure 1 describes the application of M&S to support verification and validation of more detailed system alternatives. This wide range of intended applications is consistent with the general definition of systems engineering, as presented in INCOSE (2015), where systems engineering is defined as "an interdisciplinary approach and means to enable the realization of successful systems …([focused] on defining customer needs and required functionality early in the development cycle, documenting requirements, and then proceeding with design synthesis and system validation while considering the complete problem" (pp. 11). This holistic approach to problem solving is reflected in many systems engineering process models, which are foundational to any systems engineering effort. A comprehensive review of process modeling is beyond the scope of this project, but it is worthwhile to review several fundamental concepts described in the most popular systems engineering model, the Vee Model, prior to discussion of the role of modeling to support systems engineering. Due to its popularity, there are near countless instantiations of the Vee Model, given the scope of this project Figure 2 chooses a version presented in Prosnik (2010) that describes the Vee Model for the Defense Acquisition University.

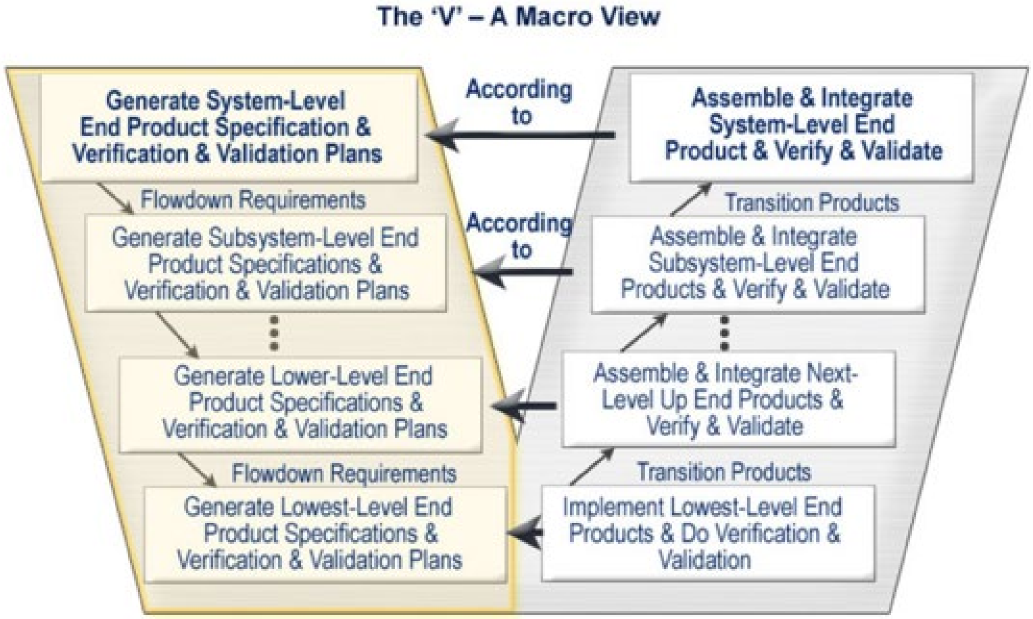


Figure 2. Systems Engineering Vee Model (Prosnik, 2010)

There are two primary takeaways from this version of the Vee Model relevant to this project. First, note that the Vee Model graphically divides the systems engineering process into a left side and a right side of the "vee." This is done to position the activities on the left side of the vee as definition and design activities focused on the translation of some ill-defined concept to a tangible system. The right side is then defined as modeling and assessment activities that support verification and validation of the correctness of the activities conducted on the left side of the Vee Model. That general structure will inform the definition of the CF-MBSE approach detailed in this project. Second, note that the specific activities described as a user moves from left to right through the design process reflect the INCOSE definition of systems engineering presented earlier. Generally, the Vee Model depicts a process where a user defines the concept for a system of interest and subsequently translates that definition, through a series of progressively more detailed design decompositions, into a model of a candidate system configuration (the definition of which corresponds with the bottom of the vee). As the user continues employing the model, those models of candidate system configuration are assessed and implemented. This general structure of system definition, system design, system modeling, system analysis, and system implementation will also inform the definition of the CF-MBSE approach presented in this project.

Beyond this general description of systems engineering concepts and process models, there has been substantial work done in the past 20 years that influence the efforts conducted in this project. Specifically, there is substantial overlap between the intended application of M&S to support systems engineering and the overall goals of systems engineering as a discipline. This overlap has resulted in a recent emphasis on the definition and development of a new field within systems engineering, termed model-based systems engineering (MBSE).

Model-Based Systems Engineering

Within INCOSE (2015), MBSE is contrasted with document centric systems engineering and defined as “the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases” (pp. 189). Note the similarity between this definition and the general definition of systems engineering. While the overall goal remains the same (system development and management throughout the life cycle), MBSE specifies the process through which that development and management will occur, namely that a formalized application of modeling will be used as the approach and means to support the system.

To expand that general definition, Friedenthal et al. (2007) formally define five intended benefits of MBSE. First, MBSE improves communication among stakeholders by establishing an authoritative system model. Second, MBSE increases the ability to manage system complexity by standardizing modeling representations and thereby allowing a single model to present the same system from multiple perspectives. Third, MBSE improves product quality through definition of an unambiguous system model. Fourth, MBSE enhances knowledge capture by standardizing data elements to reduce the cost (both financial and time commitment) of implementing changes to system design. Finally, MBSE improves the ability to teach and learn SE fundamentals by clarifying modeling approaches and concepts. Those goals are expanded by Estefan (2008), who surveys candidate MBSE methodologies and provides instruction regarding the implementation of MBSE to support engineering efforts.

Additional efforts over the past 10 years have expanded that initial conceptualization of MBSE. Both Hart (2015) and Huldt and Stenius (2019) provide descriptions of MBSE fundamentals and the evolution of the field. Dickerson and Mavris (2013) provide a summary of the fundamental concepts of MBSE and demonstrate an application to improve decision making through improved system relationships. Ryan et al. (2013) provide a roadmap for the utilization of MBSE products to support requirements engineering. Kapos et al. (2014) develop an approach for automating system analysis based on standardized MBSE products. Similarly, LaSorda et al. (2018) demonstrate that a satellite architecture can be investigated and simulated using MBSE products as a starting point. German and Rhodes (2016) and Reid and Rhodes (2016) investigate non-technical considerations for MBSE and provide insight into how models are perceived and interpreted by human decision-makers and how that perception and interpretation impacts system design decisions. Integrating these concepts, Gold (2016) suggests that a properly executed MBSE effort may serve as the starting point for developing and assessing DoD systems focused on mission capability, rather than system design characteristics. These developments, which expand the definition of MBSE from a field that aids traditional systems engineering efforts to one that serves as a starting point for more holistic operational capability analysis, influences the approach presented in this paper.

To realize the intended benefits of MBSE in support of a capabilities focused approach, two underlying themes must be clarified. First, MBSE requires that system capabilities, requirements, functions, and components can be represented through models. The specific form of these models is not specified in any MBSE standard or program; rather the intent of this approach is to eliminate the hardcopy, paper-based approach utilized in traditional SE and replace those artifacts with dynamically linked models. In a practical sense, MBSE attempts to provide systems engineers a dynamic tool to represent engineering products, thereby overcoming the traditional limitations caused by developing and presenting static diagrams developed in a program like PowerPoint (or some other non-dynamic program) in lieu of a true modeling tool. Second, MBSE requires that models can be used to accurately represent the behavior of the true system. This ensures that these models can be used to investigate the impact of component change on system performance, a challenge particularly relevant to assessment of operational capability early in the system life cycle.

The ability to properly identify, document, and assess capabilities and requirements is the basis of an MBSE-focused project. While it is impossible to create models that completely capture the characteristics and behavior of physically constructed and complete systems, MBSE focuses on creating surrogates for physical systems that can be used as the basis for analysis. While this inability to achieve complete model accuracy is a limitation, achieving a sufficient level of accuracy necessary to support assessment of operational capability through MBSE is a realistic goal. Recent advances in cluster computing, simulation development, and experimental design methodologies have greatly increased the ability of modeling and simulation to accurately represent system behavior. These enabling simulations and analysis techniques have spurred the growth of MBSE as a discipline. However, this growth has resulted in a widespread usage of MBSE technology with relatively limited attempts to examine the appropriateness of these applications. Therefore, a critical examination of the utilization of MBSE-enabled operational capability modeling is the focus of this work.

Systems Modeling Language (SysML)

Prior to discussion of operational capability modeling, it is necessary to present an overview of the Systems Modeling Language (SysML) and its relationship with MBSE. SysML was developed as an extension of the Unified Modeling Language (UML), which was established as a modeling language for software development. Widespread acceptance for UML resulted in INCOSE categorizing UML as the standard language for systems engineering in 2001 (Weilkiens, 2008). As systems engineers desired extensions to UML, the Object Management Group (OMG) expanded UML to a systems modeling language, OMG SysML, in 2006 (Weilkiens, 2008). OMG SysML is designed to be used in conjunction with UML as the standard language for systems engineering, thereby allowing independently developed models to interface seamlessly. This interoperability is a byproduct of standardized terminology for hardware, software, processes, diagrams, and interfaces established by SysML.

Over the past 20 years, SysML/UML has progressed to become the current industry language standard. A comprehensive overview of SysML and its utility as the foundation for MBSE can be found in Delligatti (2014) and Friedenthal et al. (2009). For the purposes of this project, a brief introduction to SysML is presented. Figure 3 presents a graphical description of the core views described by SysML (Object Management Group, 2006).

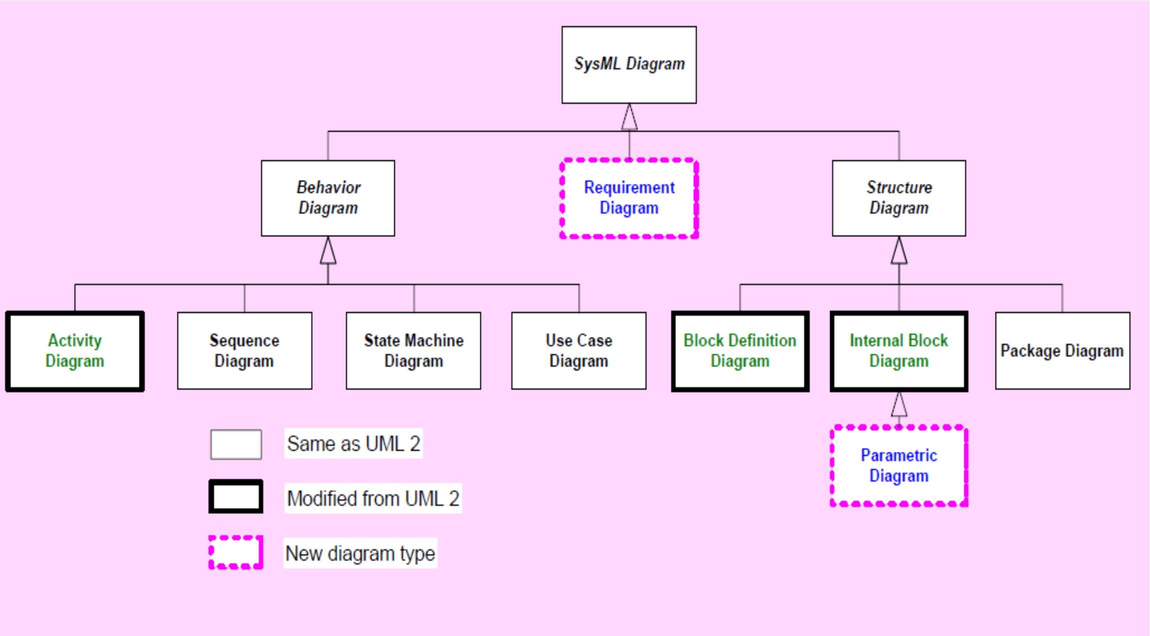


Figure 3 SysML Taxonomy and Relationship to UML (Object Management Group, 2006)

Note that while SysML retains the core concept of decoupling system behavior from system structure, as defined in UML, it adds the additional perspective of system requirements as a unique class of diagram. This is consistent with the intent of systems engineering, specifically that it is a process that supports complete system examination and therefore requires a standardized and shared vision of system requirements. Crucially, the SysML standard emphasizes that there is no mandatory starting point for the generation of SysML diagrams; the order of creation is dependent on the specifics of the project. Accordingly, a new project may follow a somewhat idealized sequence where system requirements are captured in a Requirement Diagram, which are then translated to a solution agnostic representation of system functionality in multiple Behavior Diagrams, which are ultimately described in terms of physical form in Structure Diagrams. Alternatively, a project that utilizes SysML to describe an existing system may employ a reverse engineering process, where the SysML Structure Diagrams are used to describe an existing system, are subsequently expanded into Behavior Diagrams that describe the functionality of each component, and finally are integrated with a Requirements Diagram that is fed back to stakeholders and customers for verification and validation of the existing design. Neither process is inherently more correct, for the purposes of this project the flexibility offered by SysML is particularly useful for decision making. To that end, recent work by Bleakley et al. (2011) and Russell (2012) demonstrate that SysML products may serve as a useful starting point for the conduct of trade studies. That flexibility in application is particularly useful when SysML is evaluated in the context of adherence to the standards established by the Department of Defense Architecture Framework (DoDAF).

Department of Defense Architecture Framework (DoDAF)

The DoDAF defines a set of 52 architectural views that describe systems from multiple perspectives. Those 52 viewpoints are organized into eight general categories. Figure 4 (Department of Defense Chief Information Officer, 2015) provides a graphical representation of the relationships between the DoDAF views.

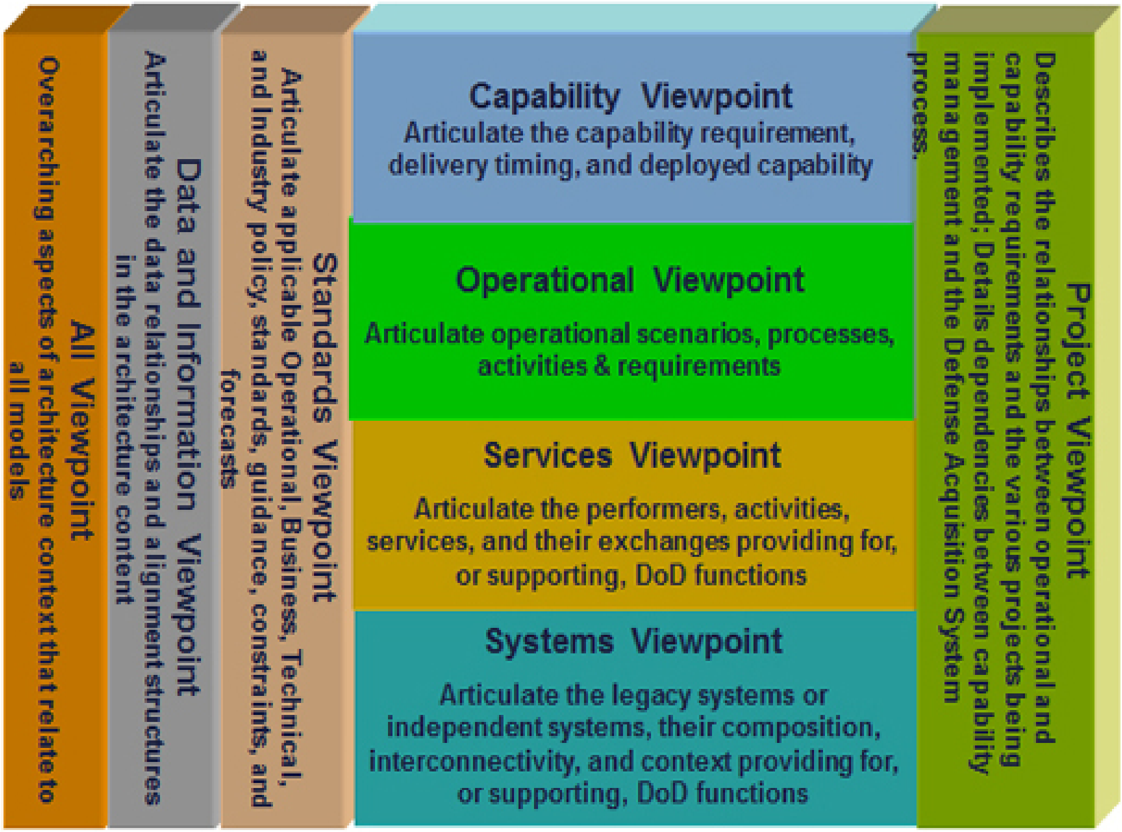


Figure 4. Department of Defense Architecture Framework (DoDAF) Viewpoints (Department of Defense Chief Information Officer, 2015)

Note that there are four core viewpoints defined by DoDAF: the Capability Viewpoint, the Operational Viewpoint, the Services Viewpoint, and the Systems Viewpoint. These viewpoints are intentionally arranged hierarchically, where capabilities describe more detailed operations, which require specific services to be performed, which utilize individual systems. This hierarchical structure notionally ensures that any system or component described within the Systems Viewpoint can be traced up to a delivered capability. The DoDAF also specifies four additional perspectives (the Project Viewpoint, the Standards Viewpoint, the Data and Information Viewpoint, and the All Viewpoint) which span this hierarchical structure and necessarily interact with models or views at multiple levels. Crucially, the DoDAF also defines an underlying data structure, where the specific elements that are necessary for the creation of each viewpoint are defined and the relationship between those elements is described. Much like SysML, the DoDAF also offers tremendous flexibility regarding the creation of each viewpoint. As an example, depending on the specific modeling approach and decisions, a SysML Activity Diagram and a SysML Sequence Diagram may both be compliant with the standards for a DoDAF OV-5b. The flexibility offered by both the DoDAF and SysML, coupled with the unambiguous definitions of the underlying data elements, makes them a suitable foundation for the development of the CF-MBSE approach. The relationship of the models developed in each of the methodologies can be defined per SysML guidelines. SysML allows for specification of an underlying schema where “component models” are subordinate to “system models,” which in turn are subordinate to “operational models” and “capability models.” This allows for consistency with the DoDAF structure. This also enables an MBSE approach to establish a defined traceability between individual system components, operational need, and system capabilities.

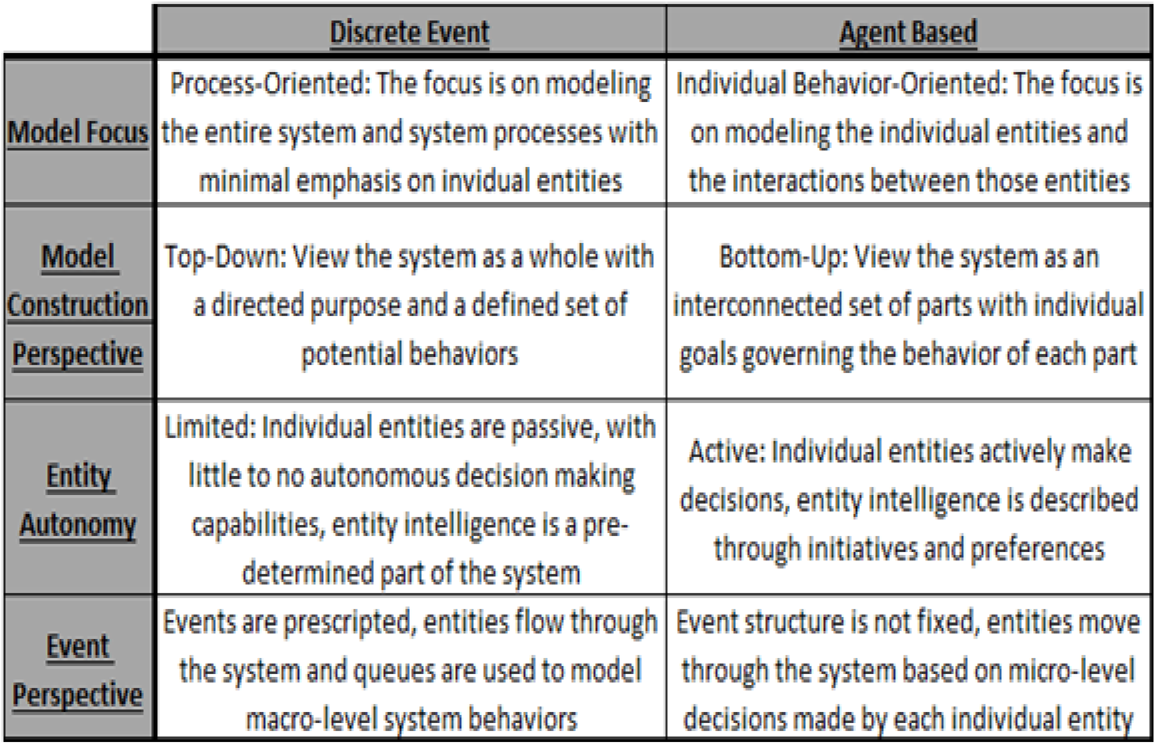
Operational Capability Assessment

In order to expand the conceptualization of MBSE efforts from an architectural focus to one that utilizes the capabilities of operational simulation models it is necessary to categorize and review different types of models and discuss metric development.

Model Selection

The term *model* is broad; this project is specifically interested in the utilization of computer-based constructive simulation models to support operational capability assessment consistent with previously developed architecture products. Accordingly, both discrete event and agent-based simulations are appropriate to support analysis within the context of the CF-MBSE Framework. Generally, discrete event simulations are recommended for systems whose behavior is defined by a series of events. For systems where interactions with other systems (or the external environment) are of particular interest and the exact behavior of the system cannot (or should not) be defined explicitly, agent-based models are recommended. A concise recommendation regarding the use of discrete event and agent-based simulations to assess system of system performance is presented in Baldwin et al. (2015), who state that discrete event simulation is recommended when “examining the results of a system” and agent-based simulation is recommended when “the modeler is interested in characteristic behavior of the system of interest rather than the results of a system activity.” For reference, Table 1 provides a concise summary of the limitations and advantages of both agent-based and discrete event simulation and is intended to provide guidance to any user who is attempting to decide between using agent-based or discrete event simulation in the context of the CF-MBSE Framework. This table integrates work from several sources, including Siebers et al. (2010) and Behdani (2012) who present a clear overview of the trade-offs between the different model types. As a note, this project rejects several of the claims presented in those articles, specifically that discrete event simulations do not represent the external environment (while they do not represent environment explicitly, the environment can be modeled implicitly), that discrete event simulations do not represent component interactions, and that agent-based simulations are based on theory and subjective data (proper development of agent-based simulations, particularly when establishing system design parameters, can and should be based on measured, objective data). As long as these limitations are acknowledged and understood, Table 1 should provide sufficient detail regarding the model focus (process vs. behavior oriented), model construction (top down vs. bottom up), entity autonomy (limited vs. active), and event perspective (pre-scripted vs. individual decisions) to guide a user to a proper simulation model. Note that Table 1 is not intended to provide a comprehensive definition of discrete event and agent-based simulations, rather it is solely intended to provide a concise set of directions.

Table 1. Comparison of Discrete Event and Agent-Based Modeling Approaches



	Discrete Event	Agent Based
Model Focus	Process-Oriented: The focus is on modeling the entire system and system processes with minimal emphasis on invidual entities	Individual Behavior-Oriented: The focus is on modeling the individual entities and the interactions between those entities
Model Construction Perspective	Top-Down: View the system as a whole with a directed purpose and a defined set of potential behaviors	Bottom-Up: View the system as an interconnected set of parts with individual goals governing the behavior of each part
Entity Autonomy	Limited: Individual entities are passive, with little to no autonomous decision making capabilities, entity intelligence is a pre-determined part of the system	Active: Individual entities actively make decisions, entity intelligence is described through initiatives and preferences
Event Perspective	Events are prescripted, entities flow through the system and queues are used to model macro-level system behaviors	Event structure is not fixed, entities move through the system based on micro-level decisions made by each individual entity

Metrics Development and Evaluation

Metrics development and evaluation, within the context of an MBSE effort, is a nontrivial problem. As with any systems engineering or acquisition effort, there exist subtle differences in how metrics are developed and solution evaluation is conducted. Rather than describe these differences in detail, a representative text is chosen as a baseline, and a generalized definition is subsequently proposed. Buede (2009) states, "Evaluation should reveal which of several design alternatives is preferred" (pp. 185). This definition clearly and concisely states the objective of solution evaluation. However, in order to establish which design (i.e., solution) alternative is preferred, metrics that facilitate this evaluation must first be defined.

Buede (2009) expands his discussion of solution evaluation by pulling from the INCOSE pragmatic principles as a basis for defining measures of effectiveness (MOEs) and measures of performance (MOPs). Rather than use Buede's definitions or attempt to develop generalized definitions, INCOSE's definitions of MOEs and MOPs can be used for the duration of this research. INCOSE defines MOEs as "the 'operational' measures of success that are closely related to the achievement of the mission or operational objective being evaluated, in the intended operational environment under a specified set of conditions; i.e., how well the solution achieves the intended purpose" (INCOSE, 2015, pp. 133). MOPs are defined as "the measures that characterize the physical or functional attributes relating to the system operation, measured or estimated under specified testing and/or operational environment conditions" (INCOSE, 2015, pp. 133). INCOSE expands these definitions with two critical clarifying points. MOEs are "the overall operational success criteria" and "MOPs are used to assess whether the system meets design or performance requirements" (INCOSE, 2015, pp. 134). This expansion clarifies the critical difference between an MOE (operational success) and an MOP (performance requirement). These definitions successfully define MOEs and MOPs to the point that any systems engineer should be able to apply them to any system of interest in order to perform system/solution evaluation. However, in order to correctly evaluate the performance of a system with regard to an MOE or an MOP, some sort of desired or expected system behavior must be established as a comparison point for evaluation of the actual system behavior. The definition and characterization of that desired and expected behavior is the role of SysML compliant architectural representations within the CF-MBSE approach.

Linkage to CF-MBSE

The solution evaluation phase of any systems engineering process is intended to shed light on the relative performance of various potential solutions, as measured against defined MOEs and MOPs. This correct identification of these MOEs and MOPs is essential when system modeling is chosen as the solution evaluation technique. This increased importance in MOE/MOP definition for models is based on the modeling techniques typically used in systems engineering.

There is one vital point to make regarding any program or organization being examined by a rigorous systems engineering process. The application of such a process is not necessary for trivial, easily understood systems. Rather, systems engineering is more appropriate applied to support the development and analysis of complicated, potentially misunderstood systems. As such, solution analysis techniques within any systems engineering process typically attempt to decompose a system into relevant subsystems, understand the behavior of those subsystems, and then represent the behavior of the entire system (or at least the behavior of the major subsystems) as an aggregated unit, typically by representing the entire system behavior as a set of mathematical models. Sequentially, the system is decomposed, understood, aggregated, and modeled. This enables the systems engineer to focus on a limited number of variables and interactions and draw potentially useful conclusions regarding the behavior of the system. This expected aggregation results in two major consequences regarding modeling of the type of complicated systems of interest to this research, specifically the analysis of the model results:

1) An extremely high level of precision will be required.
2) The data requirements will be substantial.

These consequences are potential challenges. If the representation of system component behavior is imprecise, the resultant behavior of the system may be incorrect. Also, if the behavior of each component must be represented in the system model, the data required to represent all of these components, as well as the interactions between these components, will be substantial. While modeling of complex systems is certain to require a large number of extremely precise data points, this is not a unique problem. Various methods of data analysis have developed sufficiently to allow analysis of extremely large datasets. In particular this obstacle is often overcome for traditional systems through advanced design of experiments, the application and utility of which are described in detail in MacCalman et al. (2015), Sanchez and Wan (2012), and Santner et al. (2003). Recent work in the MBSE and modeling domain has demonstrated that agent-based models are also particularly useful in this area as a result of their flexibility to non-linear or non-traditional logical structures, which are generally a prerequisite to the utilization of discrete event simulation models. Notably, both Acheson et al. (2013) and Maheshwari et al. (2018) demonstrate a linkage from MBSE concepts to agent-based models. Accordingly, and in recognition of the unique ability of agent-based models to support assessment of system behavior when the exact system structure may not be known, this research focuses on development of a CF-MBSE approach with a demonstration of operational capability assessment using agent-based modeling.

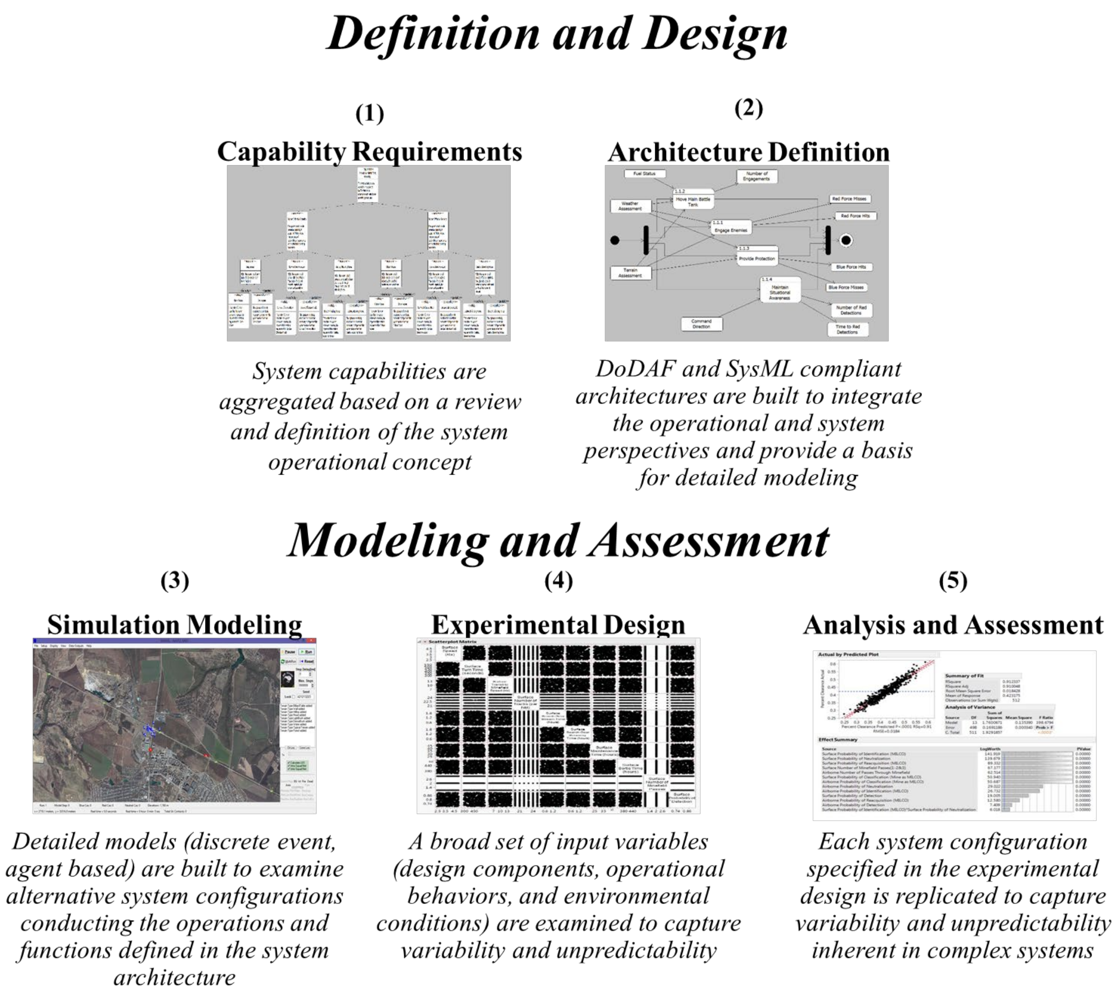
THIS PAGE INTENTIONALLY LEFT BLANK

Capabilities Focused Model-Based Systems Engineering

In order to fully realize the potential of MBSE to support engineering and acquisition of new platforms and formations, a CF-MBSE Framework is proposed to establish a clear linkage between the architectural representations defined in SysML and operational modeling approaches. This allows for an expansion of the conceptualization of MBSE in such a way that facilitates comprehensive examination of potential system configurations. In that way, the CF-MBSE approach establishes a framework that facilitates the application of INCOSE's intended benefits of MBSE to the analysis and acquisition of future systems.

CF-MBSE Framework Description

As discussed in the MBSE overview, substantial recent work has been done on development of systems engineering artifacts. Given that the intended application of the CF-MBSE framework is to support engineering and acquisition of DoD systems, both SysML products and DoDAF viewpoints are specified within the approach. To ensure applicability, the CF-MBSE framework is designed to integrate those standard architectural representation to support more detailed modeling that can support system acquisition. This should ensure that the framework is applicable to both the general systems engineering and system development community and provide additional impact to the DoD community. An overview of the approach is shown in Figure 5, implementing the previously described division of systems engineering efforts into system definition and design and system modeling and assessment.



System capabilities are aggregated based on a review and definition of the system operational concept

DoDAF and SysML compliant architectures are built to integrate the operational and system perspectives and provide a basis for detailed modeling

Detailed models (discrete event, agent based) are built to examine alternative system configurations conducting the operations and functions defined in the system architecture

A broad set of input variables (design components, operational behaviors, and environmental conditions) are examined to capture variability and unpredictability

Each system configuration specified in the experimental design is replicated to capture variability and unpredictability inherent in complex systems

Figure 5. CF-MBSE Framework

Definition and Design

Figure 5 aligns with the general systems engineering process outlined earlier of system definition, system design, system modeling, system analysis, and system implementation (with the note that system implementation is currently beyond the scope of this research). Figure 5 briefly expands each step of that generalized process. System Definition is focused on identification of Capability Requirements. This term intentionally mixes the terminology championed by DoDAF as the starting point for system definition (capabilities) and the de facto term used to start many leading systems engineering processes (requirements). Note that this step also comprises review and definition of an operational concept. While it may notionally be possible to describe and define a system in an operationally agnostic form, the CF-MBSE approach is presented to guide the engineering and assessment of systems that provide a specific capability. Given the definition of capability presented in (Department of Defense Chief Information Officer, 2015) as “the ability to achieve a Desired Effect under specified (performance) standards and conditions through combinations of ways and means (activities and resources) to perform a set of activities,” it is appropriate to assume that definition of a relevant operational concept is necessary to the identification of the desired effect that the system is expected to provide.

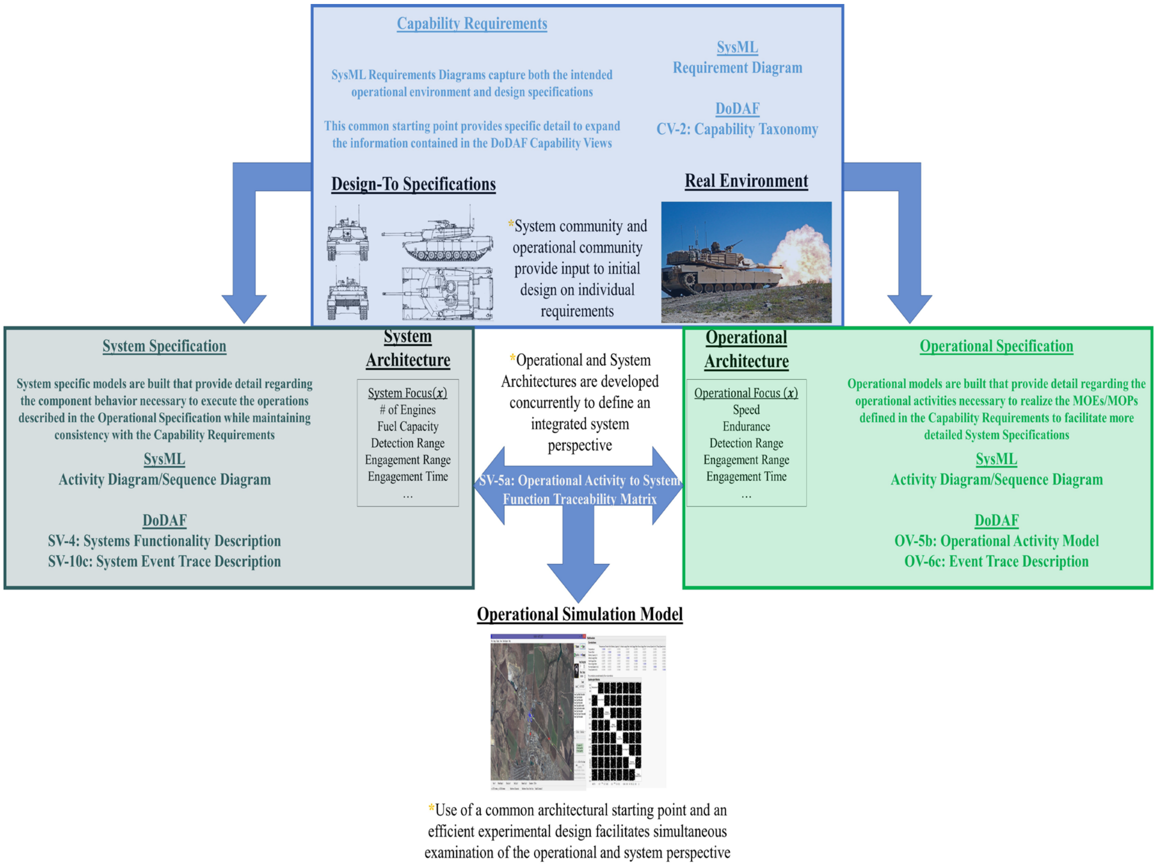
Accordingly, the first step in the CF-MBSE framework is the generation of a SysML Requirement Diagram that describes the capabilities that the system of interest is expected to provide in its operating environment. Notably, this is consistent with the goals and aims of the DoDAF capability viewpoint, specifically the DoDAF CV-2: Capability Taxonomy. The DoDAF CV-2 is a hierarchical representation of system capabilities, in particular those capabilities that assist in definition of user requirements and high-level use cases. Rather than define or develop a new underlying relationship between requirements and capabilities the CF-MBSE advocates using the term *capability requirement* consistent with the DoDAF definition of system capabilities to describe the specific system characteristics generally associated with system requirements. This simplifies the initial system definition process and provides the additional benefit of allowing the use flexibility to associated MOEs and MOPs freely, without concern for assignment to a specific term.

As a note, an alternative approach could define requirements and capabilities as distinct model data types tied together with an appropriate relationship (for example, requirements “provide” capabilities or requirements “implement” capabilities). While such an approach may provide value to some users (and is the approach utilized in the MBSE software used to generate architectural representations in this project, Vitech CORE), definition of such a relationship complicates the initial system definition process and does not serve the primary goal of identification of Capability Requirements, namely that they establish a starting point for the system Architecture Development (step 2 of the CF-MBSE approach).

After Capability Requirements have been developed that describe the high-level utilization of the system of interest, they are used to the bound the creation of architectural products that describe the operations of the system of interest as well as the resources that the system requires to execute those operations. The CF-MBSE approach advocates adhering to the DoDAF definition and intended utility of the Operational Viewpoint and the Systems Viewpoint to support this architectural modeling.

The DoDAF defines the Operational Viewpoint as a perspective that "describes the tasks and activities, operational elements, and resource flow exchanges required to conduct operations" (Department of Defense Chief Information Officer, 2015). Of note, the Operational Viewpoint intentionally avoids specification of a material solution that implements the tasks and activities described in each of the associated models. This is done to avoid driving towards a preferred physical configuration and instead focus on system behavior without consideration for a specific system component or element. Accordingly, the CF-MBSE framework advocates initiating the Architecture Development process with the construction of either a DoDAF OV-5b: Operational Activity Model or DoDAF OV-6c: Event Trace Description. In practice, the two models should be interchangeable, the selection of one or the other is left to user preference and expertise. Note that each of these models can be created in compliance with SysML using Behavioral Diagrams. The DoDAF OV-5b identifies the SysML Activity Diagram as compliant with the DoDAF standard and the DoDAF OV-6c suggests a SysML Sequence Diagram as an approach to meeting the DoDAF standard. Creation of either (or both) of these diagrams defines the system in a solution agnostic form that is linked directly to the Capability Requirements developed in the previous step. To add specificity to the architectural model, the DoDAF System Viewpoint can be used as a follow on to the Operational Viewpoints within the Architecture Development step of the CF-MBSE approach.

DoDAF defines the Systems Viewpoint as a perspective that "describes systems and interconnections providing for, or supporting, DoD functions" (Department of Defense Chief Information Officer, 2015). Crucially, the Systems Viewpoint identifies the physical resources that are required to support and execute the activities described in the Operational Viewpoint. For the CF-MBSE approach, the DoDAF SV-4: System Functionality Description and the DoDAF SV-10c: System Event Trace Description are particularly useful. As with the Operational Viewpoints, DoDAF identifies SysML diagrams that are compliant with both the SV-4 and the SV-10c. Once again, the SysML Activity Diagram is compliant with one view (the SV-4) and the SysML Sequence Diagram is compliant with one view (the SV-10c). This similarity allows for linkage of the Operational Viewpoint and System Viewpoint. Both viewpoints trace directly from the Capability Requirements presented in the CV-2 and describe the intended behavior of the system of interest, with the sole difference that the Operational Viewpoint is material solution agnostic and the Systems Viewpoint is material solution specific. Figure 6 provides a graphic expansion of the definition and design steps of the CF-MBSE approach.



SysML Requirements Diagrams capture both the intended operational environment and design specifications

This common starting point provides specific detail to expand the information contained in the DoDAF Capability Views

*System community and operational community provide input to initial design on individual requirements

System specific models are built that provide detail regarding the component behavior necessary to execute the operations described in the Operational Specification while maintaining consistency with the Capability Requirements

*Operational and System Architectures are developed concurrently to define an integrated system perspective

Operational models are built that provide detail regarding the operational activities necessary to realize the MOEs/MOPs defined in the Capability Requirements to facilitate more detailed System Specifications

*Use of a common architectural starting point and an efficient experimental design facilitates simultaneous examination of the operational and system perspective

Figure 6. Linkage of Capability Requirements and Architecture Development for CF-MBSE

Figure 6 stresses three important points. First, both the system community and the operational community will likely begin the engineering and acquisition effort with differing terminology and perspective. The intent of developing the DoDAF CV-2 as the starting point for the CF-MBSE approach is to standardize terminology, thinking, and perspective prior to more detailed system development. Second, the Architecture Development step of the CF-MBSE approach is divided into two efforts, the Operational Architecture and the System Architecture. These architectures are developed through generation of the DoDAF products described previously. Finally, note that an additional DoDAF product, the SV-5: Operational Activity to System Function Traceability Matrix is suggested. This is not mandatory, but creation of a traceability matrix that assesses consistency between the operational and system viewpoints is often a worthwhile starting point for the development of more detailed simulation models.

Modeling and Assessment

As discussed in the previous chapter, both discrete event and agent-based models may be appropriate for the early identification of desirable system characteristics based on system architecture products. For the interested reader, Law (2009) provides a succinct summary of best practices for simulation model development that is expanded in Law (2014). Rather than provide an overview of the fundamentals of each class of simulation model, this project presents several comments on development of agent-based models which, given the focus on investigating a potentially wide range of system operational and design alternatives to achieve a capability, are likely the preferred modeling approach for users of the CF-MBSE approach.

Agent-based computer simulations orient model development around the behavior of each of the simulation entities. Each entity is autonomous and defined in terms its individual characteristics as well as its personality and interactions with other autonomous simulation entities. Based on the actions and interactions of these entities, changes in system behavior are observed. They are necessarily capable of modeling both intricate processes and decision logic, thereby allowing for representation of extremely precise entity/component behavior. Further, many agent-based computer simulations are library-based in nature, thereby allowing large batch runs, which enables examination of extremely large decision and solution spaces.

Note that these simulations do not remove the possibility of misrepresenting the overall behavior of the system. For any complicated system being studied with the CF-MBSE approach, orienting it as an agent-based simulation only removes the possibility of obtaining erroneous results based on a misrepresentation of the system itself. However, if the behavior of the simulation entities or the definition of the interactions between these entities does not accurately represent the behavior and relationships of the actual system components, then error is introduced into any results, and therefore any analysis. These simulations can only provide precise results regarding system behavior if the definition of the simulation entities is accurate.

THIS PAGE INTENTIONALLY LEFT BLANK

CF-MBSE Demonstration

To demonstrate the applicability of the CF-MBSE framework, a candidate system is chosen for examination. To avoid classification or conflict, the demonstration focuses on operational assessment of a theoretical NATO main battle tank. The characteristics and operational concept for the system were developed via informal discussions with multiple individuals at the former U.S. Army Tank Automotive Research, Development, and Engineering Center (TARDEC), now the Combat Capabilities Development Command (CCDC) Ground Vehicle Systems Center (GVSC). Those discussions are used to bound the operational scenario and provide a starting point for the conceptualization of the system of interest. The architectural products and operational simulation model utilized in this paper are based on the graduate thesis research of Fernandez et al. (2018).

Capability Development

Prior to development of architectural representation of the system requirements and capabilities, an operational scenario is developed. This operational scenario will provide a boundary for definition of system capability requirements as well as MOEs and MOPs. Given that the system design recommendations from the approach will necessarily be sensitive to the selection of the operational environment, two scenarios are proposed within the operational concept. In the first scenario, the system of interest is evaluated in terms of utility and support for a defensive operation. In the second scenario, the system of interest is evaluated in an offensive operation. Per the intended use of the system as a NATO main battle tank, the geographical characteristics are representative of a notional area within the NATO area of responsibility.

Defensive Scenario

A defensive scenario is proposed where a NATO-controlled checkpoint is established along a major road to inspect transiters and control east to west movement. Within the scenario, a unit utilizing the NATO main battle tank is supporting a military police (MP) inspection unit. A red force equipped with Anti-Tank Guided Missiles (ATGM) moves from east to west. Figure 7 presents a graphical overview of the defensive operational scenario.

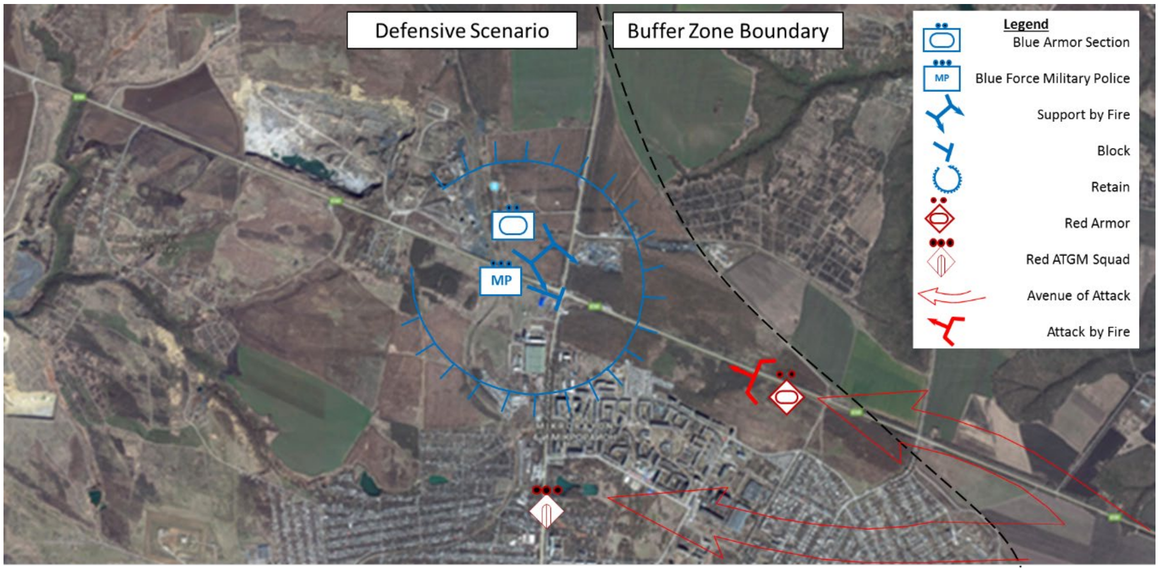


Figure 7. Defensive Scenario Graphical Description (Fernandez et al., 2018)

The behaviors of the blue force in the defensive scenario are defined in accordance with defensive operations as specified in Army Doctrine Publication (ADP) 3-90 (Department of the Army, 2012). Notably, the blue force is primarily responsible for denying access to the red force over the duration of the mission, rather than the more offensively oriented priority to destroy the red force. The scenario begins when the red force launches an attack on the checkpoint and the blue force responds by seeking cover and initiating defensive measures. The scenario has end conditions that may be triggered by either the blue or red force. From the red force perspective, the scenario may end when the blue force is completely destroyed or the red force takes control of an agent positioned in the checkpoint. From the blue force perspective, the scenario may end when the red force is completely destroyed or the model timer reaches a predetermined end time, representing the duration necessary to defend the checkpoint for a reserve force to arrive.

The MP element is managing the traffic control point, and the main battle tank section occupies a defensive battle position. The main battle tank section consists of four main battle tanks reinforced with 10 light infantry personnel. These forces are oriented defensively with the objective of discouraging any red force progress through the checkpoint. As a note, this is roughly equivalent to the definition of a tank platoon as described in FM 17-15 Tank Platoon (Headquarters Department of the Army, 1996) and shown in Figure 8.

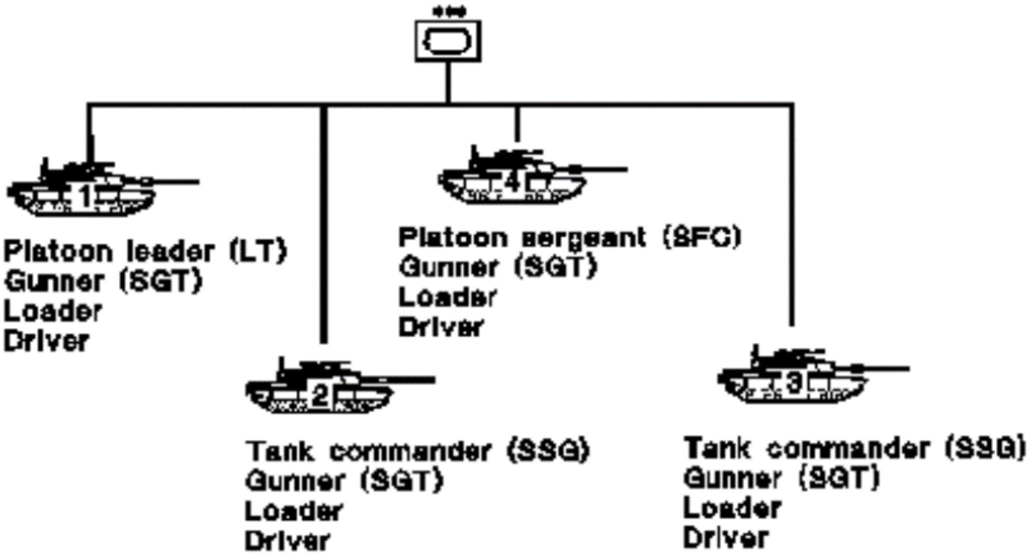


Figure 8. Tank Platoon Organization (Headquarters Department of the Army, 1996)

The red force is modeled to represent a collection of insurgent forces who may pose a realistic threat in the NATO area of responsibility. Accordingly, they are equipped with anti-tank guided missiles utilized by the red force infantry personnel. In total, the red force is comprised of 30 infantry personnel and a tank platoon of four main battle tanks.

Offensive Scenario

An offensive scenario is developed that utilizes the same geographic location as presented in the defensive scenario. The offensive scenario represents a notional response to a red force victory in the defensive scenario. The offensive scenario assumes that a red force has seized control of a NATO checkpoint in a strategically relevant location. A reserve force located 15 kilometers west of the checkpoint initiates an offensive response to regain control of the checkpoint. Figure 9 provides a graphical representation of the offensive scenario.

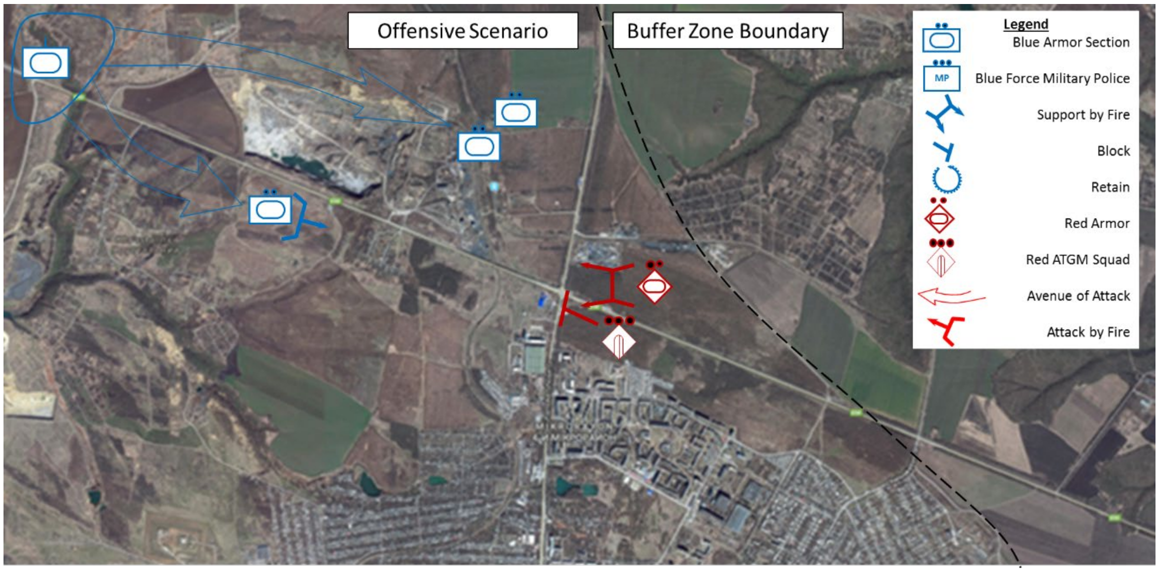


Figure 9. Offensive Scenario Graphical Description (Fernandez et al., 2018)

As with the defensive scenario, the behaviors of the blue force are based on the ADP 3-90 (Department of the Army, 2012). In this scenario, blue forces are executing an offensive task intended to both destroy enemy forces and seize terrain. Within the model, this means that the blue force continues forward movement until the red force is completely destroyed or retreats. The blue force is restricted from pursuing additional offensive action in the event of a red force retreat to mimic reinforcement of the checkpoint. Again, the simulation can be ended based on either the behavior of the red force or the blue force. The red force can end the simulation through complete destruction of the blue force or by experiencing a casualty percentage that initiates a retreat. The blue force can end the simulation through complete destruction of the red force or by capturing an agent placed within the checkpoint.

In recognition of the numerical advantage required to engage in offensive operations rather than defensive operations the blue force will be substantially larger than in the defensive scenario. The blue force is assumed to be a battalion-sized task force supported by an armor company consisting of 12 main battle tanks. This is reflective of a general operational rule of thumb where a three to one size advantage is necessary to conduct offensive operations when compared to a defensive operation in a similar environment. The main battle tanks are supplemented by light infantry personnel.

The red force is an augmented version of the red force from the defensive scenario. The assumption is that the red force has reinforced the checkpoint to mitigate any losses experienced in their attack. Additionally, the supply of ATGM is replenished and the red force is strategically positioned in areas of cover and concealment to support their defense of the checkpoint.

Capability Requirements

Per the first step of the CF-MBSE approach, the operational scenario is used as the starting point for the definition of capability requirements. For this demonstration, two high-level capabilities, support defensive scenarios and support offensive scenarios are used to aid organization. Those capabilities are implemented within a SysML Requirement Diagram and serve as a starting point for the definition of intended system capability. Figure 10 shows a graphical description of these system requirements and the first layer of decomposition beyond those capabilities. As mentioned, all MBSE architecture products are created using Vitech CORE.

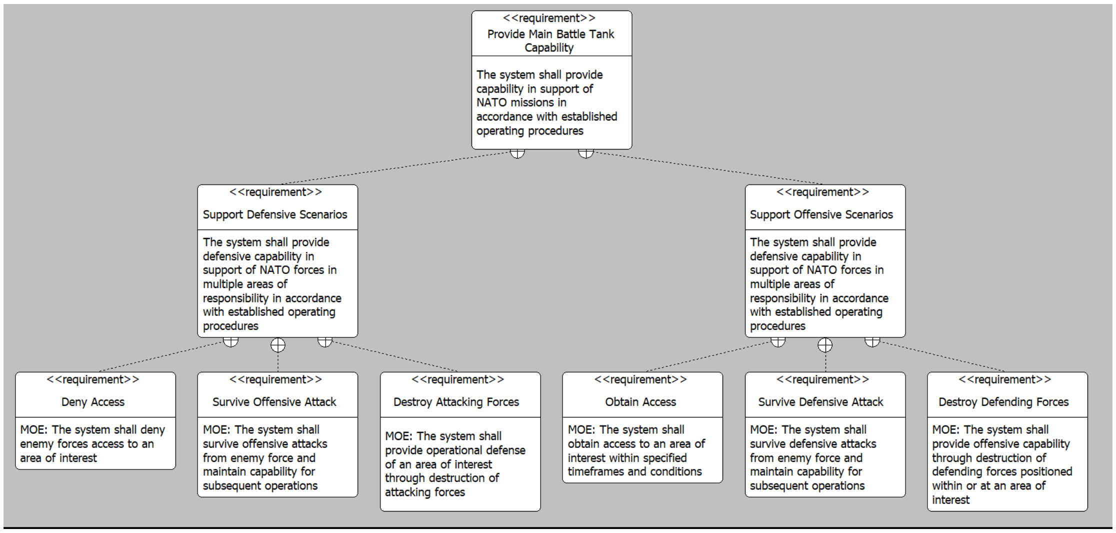


Figure 10. NATO Main Battle Tank Capabilities Implemented as SysML Requirement Diagram

Note that these diagrams will necessarily become cluttered and difficult to present in static documents; however the implementation in a modeling environment such as Vitech CORE ensures that the underlying data and relationships can be easily accessed and updated. Examination of Figure 10 shows three measures of effectiveness associated with each requirement for defensive and offensive scenarios. For the defensive scenario, the following measures of effectiveness are developed:

1. Deny Access: This MOE is the primary objective of the defensive operation, where the main battle tank is tasked with protection for the checkpoint. It is assessed in terms of red force success in obtaining control of the checkpoint.
2. Survive Offensive Attack: This MOE will necessarily be correlated with the Deny Access MOE, as zero survival of the offensive attack will result in zero ability to deny access. Note that there is an additional element to this MOE, as survival provides potential utility for follow on operations.
3. Destroy Attacking Forces: As with the Survive Offensive Attack MOE, this MOE is necessarily correlated with the Deny Access MOE, as zero ability to destroy the attacking force will result in reduced ability to deny access. This is assessed in terms of the number of red force agents destroyed in each model run.

As with the defensive scenario, there are three MOEs associated with the requirements that decompose support for offensive scenarios. Note that the offensive scenario is focused on attack of red forces and the ability of the system to gain control of the checkpoint. The MOEs associated with support for offensive operations are as follows:

1. Obtain Access: This MOE is the primary objective of the offensive operation, where the main battle tank is tasked with driving red forces from the checkpoint or destroying any red forces occupying the checkpoint. It is assessed in terms of blue force success in obtaining control of the checkpoint.

2. Survive Defensive Attack: This MOE will necessarily be correlated with the Obtain Access MOE and mirrors the “Survive Offensive Attack” MOE developed in the defensive scenario. Again, zero survival of the red force defense attack will result in zero ability to obtain access. As in the defensive scenario, there is a component of this MOE focused on support for follow on operations.
3. Destroy Defending Forces: This MOE is a straightforward assessment of the number of defending red forces destroyed by the NATO main battle tanks. This will provide an assessment of the lethality of the system and should be correlated with the Obtain Access MOE.

Beyond definition of this MOEs, a set of MOPs are developed that provides the specific data necessary to support assessment of the MOEs. Each MOP may have a varied impact on each MOE. Definition of the MOPs and assessment of their relationship to each MOE can provide additional detail regarding why the system is providing or failing to provide a capability. As mentioned, inclusion of each MOE in a static snapshot of a SysML requirement diagram is not well suited to a textual document; accordingly they are presented as a list:

1. Time to Red Detection: This MOP supports assessment of situational awareness as well as the main battle tank sensor systems. This is collected by recording the time at which the main battle tank first detects and successfully classifies a red agent.
2. Number of Red Detections: This MOP supports assessment of situational awareness as well as the main battle tank sensor systems. This is collected by recording the total number of unique red agents that the main battle tank successfully detects and classifies.
3. Red Force Misses: This MOP supports assessment of the ability of the main battle tank primary armament to successfully shoot at enemy forces. It is collected by recording the number of shots that

the main battle tank takes within a model run that miss the red force.

4. Red Force Hits: This MOP supports assessment of the ability of the main battle tank primary armament to successfully shoot at enemy forces. It is collected by recording the number of shots that the main battle tank takes within a model run that hit the red force.
5. Number of Engagements: This MOP supports assessment of the ability of the main battle tank to successfully move when engaged with enemy forces. It is collected by recording the total number of shots that the main battle tank takes within a model run.
6. Blue Force Misses: This MOP supports assessment of the ability of the main battle tank to provide protection against enemy forces. It is collected by recording the number of shots that the red force tanks take within a model run that miss the blue force.
7. Blue Force Hits: This MOP supports assessment of the ability of the main battle tank to provide protection against enemy forces. It is collected by recording the number of shots that the red force tanks take within a model run that hit the blue force.

Architectural Development

Based on the MOEs and MOPs presented in the previous section, the CF-MBSE continues with identification of the operational activities and system functions necessary to support execution of that broader capability. Those operational activities are dynamically linked within the MBSE tool to the originating requirements and MOEs. Figure 11 presents an updated version of the SysML requirements diagram showing the inclusion of the operational activities and system functions specified by each system capability requirement. The language maps directly from the system capability requirement to both the operational activities and system functions. This exact mapping supports rapid revision of modeling efforts and traceability within the MBSE tool. Note that the utility of this mapping may not be immediately apparent to individuals with limited

MBSE experience and the mapping may appear redundant. The utility of this connection will be demonstrated at the end of this section.

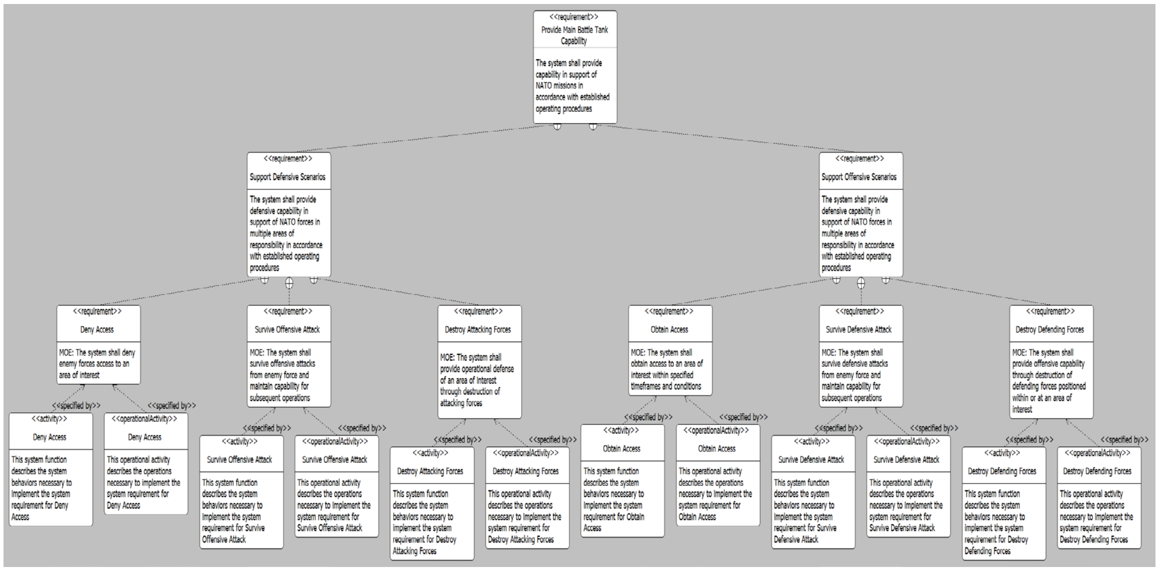


Figure 11. Expanded SysML Requirements Diagram

To that end, a SysML Activity Diagram is presented that maps the operational activities and system functions associated with the first primary capability requirement (Deny Access) in the next two sections.

Operational Perspective

Figure 12 provides a graphical representation of the Deny Access Operational Activity as a SysML Activity Diagram (note that this diagram is compliant with the DoDAF standard for the OV-5b: Operational Activity Diagram).

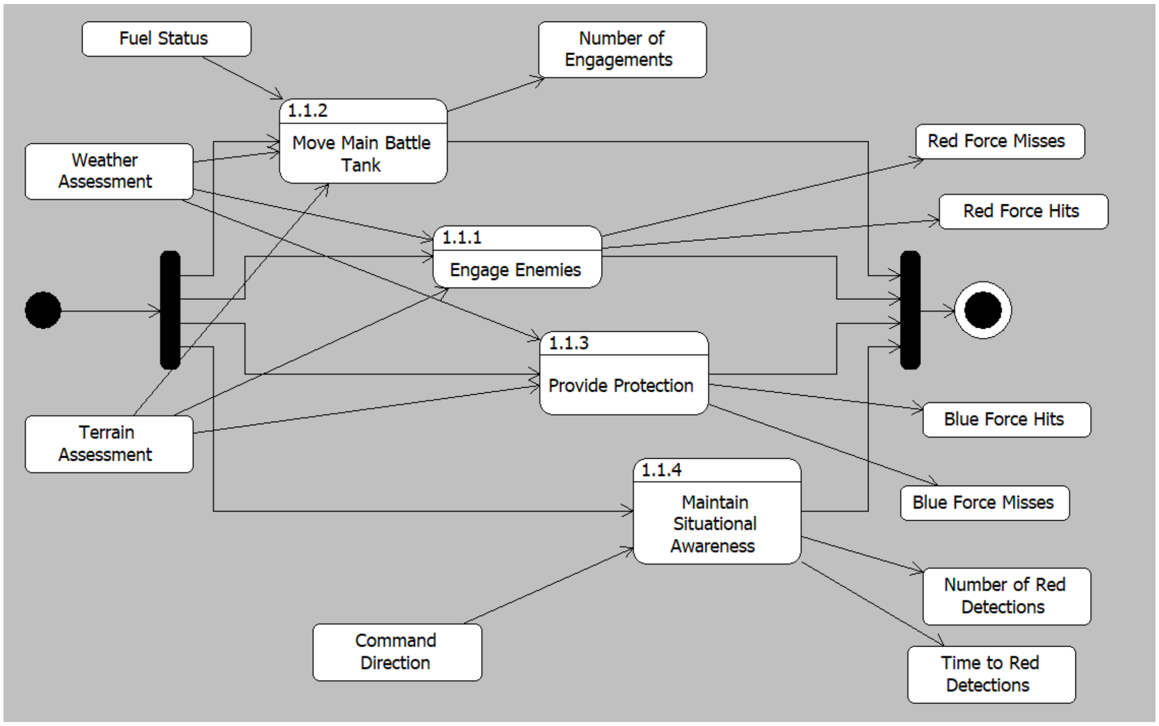


Figure 12. Deny Access OV-5b Operational Activity Model (as SysML Activity Diagram)

There are several important elements included in Figure 12 that expand the system definition beyond the initial decomposition of capabilities shown in the Requirement Diagram. First, specific operational activities that support the “Deny Access” MOE are shown. In this case, the main battle tanks conduct four operational activities: Move Main Battle Tank, Engage Enemies, Provide Protection, and Maintain Situational Awareness. Note that each of these operational activities are conducted in parallel, indicating that there is no enforced sequence of activities. As mentioned, this lack of defined sequence suggests that an agent-based modeling approach, where there is no strict ordering of operations, may be an appropriate choice for operational modeling. It is also notable that the MOPs described in the previous section are shown on the Activity Diagram. Each of the operational activities associated with Deny Access is responsible for generation of data that can be used to support assessment of each MOP. As an example, when the system conducts the Move Main Battle Tank Operational Activity, the MOP Number of Engagements is updated and recorded. Similarly, as the Engage Enemies operational activity is performed, the MOPs for Red Force Misses and Red Force Hits are generated.

There are two important benefits that result from the creation of an operational activity model using a SysML Activity Diagram. First, the diagram establishes a bridge between the high-level use case described in the Requirement Diagram and the detailed system behaviors that must be modeled in any subsequent effort. Second, development of the Activity Diagram in an MBSE tool (this project uses Vitech CORE) forces the user to define the operational activities and associated creation of inputs and outputs using standard, unambiguous terminology. In turn, this automates the creation of additional behavioral diagrams, in this case a SysML Sequence Diagram, that provides an alternative representation of the system behavior. Recall that the SysML Sequence Diagram is also compliant with the standards for a DoDAF OV-6c: Event Trace Description (Figure 13).

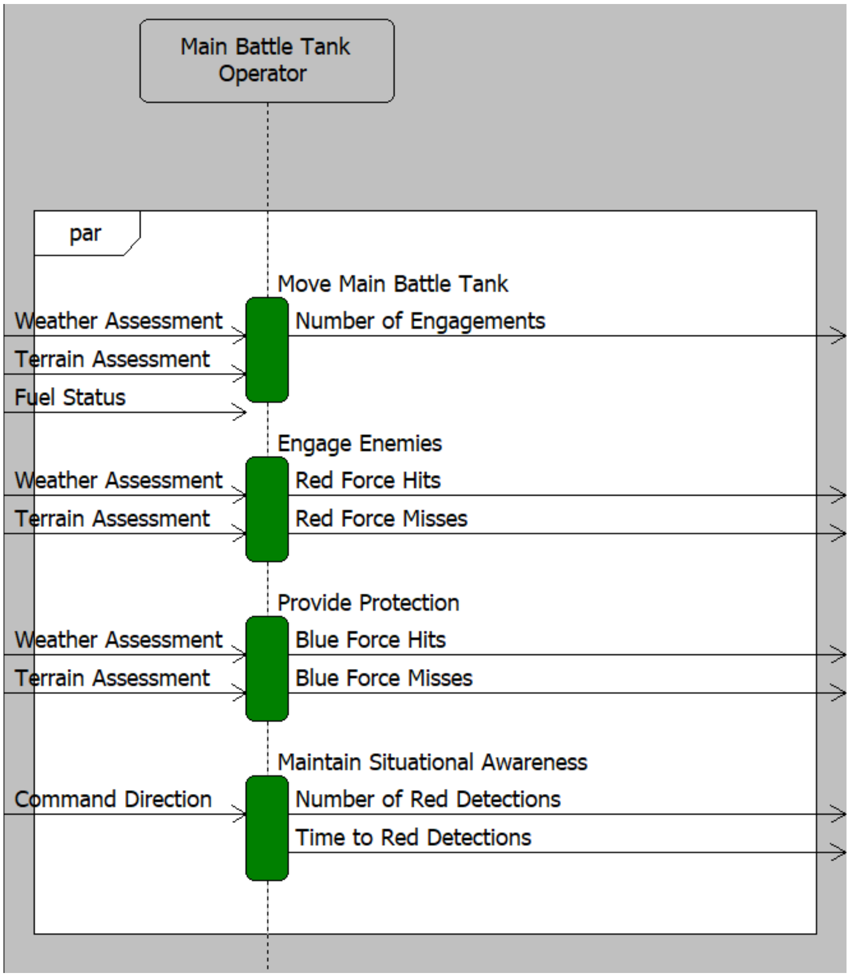


Figure 13. Deny Access OV-6c Event Trace Description (as SysML Sequence Diagram)

Note that the operational activities and associated inputs/outputs shown in the OV-6c mirror the elements shown in the OV-5b. This is a feature of Vitech CORE as an MBSE compliant architectural tool, once the model elements have been defined consistently with CORE’s underlying schema the creation of each diagram is automated by the program. Note that there is one piece of additional information provided by the Sequence Diagram that is not shown in the Activity Diagram, specifically the allocation of each operational activity to the performer responsible for execution of each activity. In this case, that allocation is straightforward: the performer Main Battle Tank Operator is responsible for each activity. In more complicated models, this allocation will become more valuable and may be a reason for choosing to develop an OV-6c rather than an OV-5b.

System Perspective

Based on the completed Operational Viewpoints and consistent with the previously developed Capability Requirements, System Viewpoints are developed that specify the system components or resources that are necessary to execute each operational activity. Note that this does not necessarily imply a one-to-one matching between the operational and system perspective. Rather, just as the operational viewpoint provided specificity regarding the activities that each performer executes to realize a capability requirement, the systems viewpoint provides specificity regarding the functions that each system executes to realize the same capability requirements. Figure 14 presents a DoDAF SV-4, built using a SysML Activity Diagram, that describes the system functionality associated with the Capability Requirement to Deny Access.

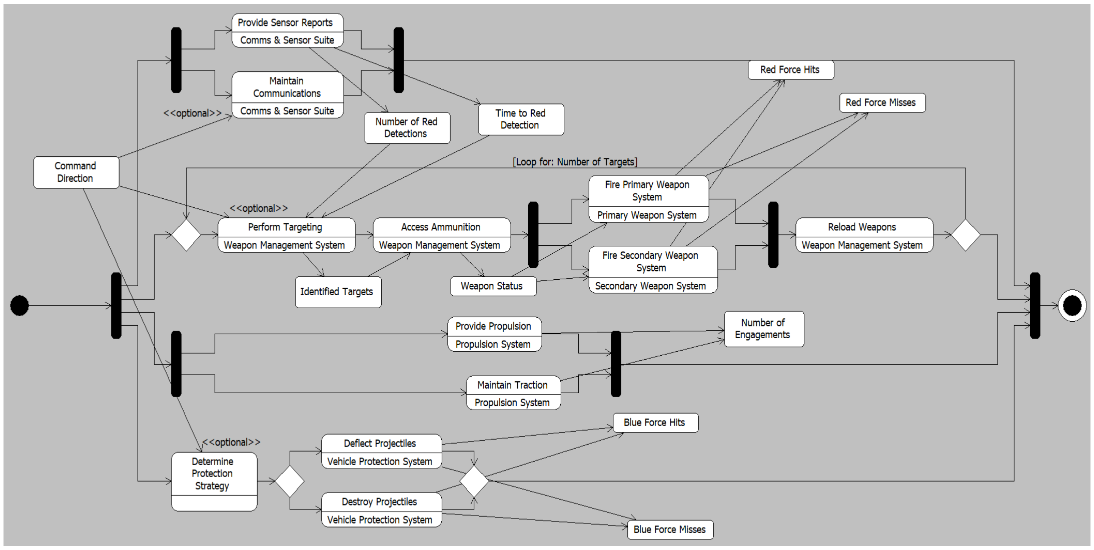


Figure 14. Deny Access SV-4 System Functionality Description (as SysML Activity Diagram)

Notice that the SV-4 is substantially more detailed than the OV-5b counterpart, despite both diagrams presenting a behavioral perspective of the Capability Requirement for Deny Access. This is a function of the specific behavior being described for the main battle tank and is a not a general rule or recommendation. In both cases, there are four sequences associated with the tank's situation awareness, engagement, movement, and protection behaviors. The system perspective provides additional detail regarding the exact functions that are necessary to represent these general behaviors in a simulation model. Note that the output from of the systems view mirrors the output of the operational view. Specifically, each process produces the data necessary to support calculation of each MOP (Number of Engagements, Red Force Hits, Red Force Misses, Blue Force Hits, Blue Force Misses, Number of Red Detections, and Time to Red Detections). As with the operational architecture perspective, the systems architecture perspective can also be represented as a Sequence Diagram (compliant with the DoDAF SV-10c) to provide additional clarity regarding the allocation of each function to system components (Figure 15).

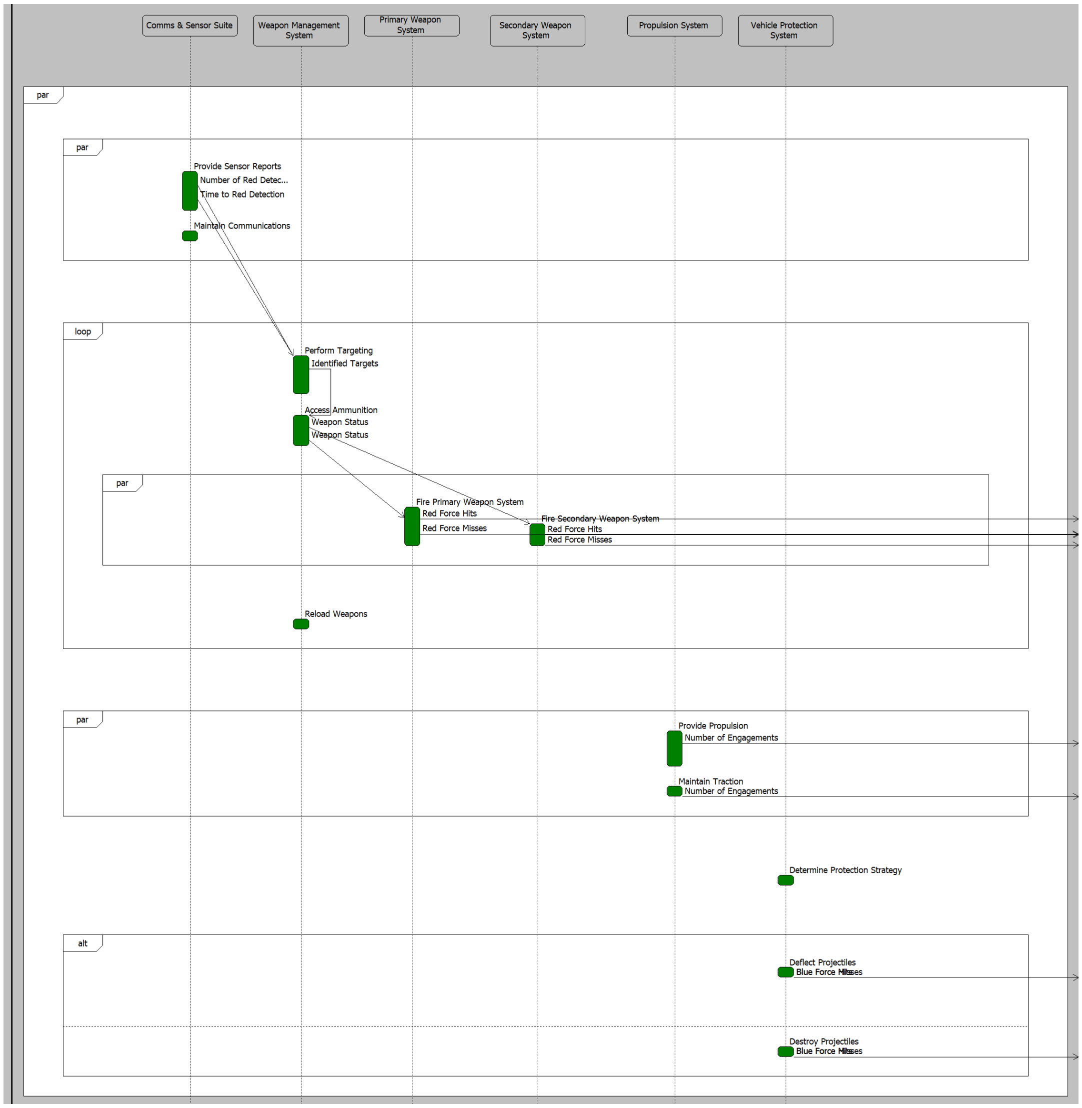


Figure 15. Deny Access SV-10c System Event Trace Description (as SysML Sequence Diagram)

As with the Activity Diagram shown in Figure 14, the system view representation of the behaviors associated with Deny Access is substantially more detailed than the operational view counterpart. Again, this is not a general rule for development of these products, and there may be situations where the operational behaviors are substantially more detailed than the system behaviors. In this specific case, the sequence of system functions is comprised of four parallel processes, and the engagement process is conducted in a loop for the number of targets that are presented to the main battle tank. Additionally, where the operational viewpoints allocated all operational activities to a single performer, the SV-10c identifies six primary system components that are responsible for execution of the system functions associated with Deny Access. These system components, and the behaviors associated with them, define the variables that should be considered as part of the operational simulation model in the third step of the CF-MBSE approach.

Simulation Modeling

As discussed, agent-based simulations are well suited to situations where the systems of interest have substantial autonomy and their behavior cannot be defined in a linear (or near linear) series of events. Accordingly, and to allow for representation of the main battle tank in a representative operational environment, an agent-based simulation called Map Aware Non-Uniform Automata-Vector (MANA-V) is selected to assess the operational performance of the main battle tank. MANA-V has been used successfully on previous NPS research projects. The program is developed and maintained by the New Zealand Defense Technology Agency (DTA) and has several desirable characteristics. First, the agents are autonomous, meaning that a general set of personalities and system characteristics can be defined prior to each model run but each agent is self-directed after the start of the simulation. Second, the program is map aware, meaning that the main battle tank can be modeled in an operational environment that reflects the potential area of system employment. Finally, MANA-V supports execution in batch mode where the number of replications for each system configuration can be specified by the user. Given that MANA-V is a stochastic model, it is necessary to replicate the model for each candidate system configuration to properly characterize the variability associated with operational performance.

The Operational and Systems Viewpoint architectural products are used as a starting point for the behaviors modeled in MANA-V. Generally, the operational viewpoints describe the overall personality of the main battle tank and the systems viewpoints provide the additional detail and allocation to system components necessary to develop an experimental design strategy that facilitates identification of preferred system configurations. Prior to review of the experimental design strategy, two important areas regarding model developed must be reviewed. First, the characteristics of the blue and red main battle tanks are established. Second, the operational environment described in the previous section is implemented in MANA-V.

Main Battle Tank Baseline Characteristics

To approximate the characteristics of the systems of interest in the operational model, the system performance characteristics from *Ground Systems: Worldwide Equipment Guide Volume 1* (Department of the Army, Training and Doctrine Command, G-2, 2016) are utilized. Specifically, the red force main battle tank is modeled after the T-72B3, and the NATO main battle tank is modeled as an enhanced combination of the Challenger 2 and the Leopard 2, as defined in Department of the Army, Training and Doctrine Command, G-2 (2016). Additionally, an external armor protection system is considered as a potential design expansion for the NATO main battle tank (this external armor protection system is not implemented on any versions of the red main battle tank). Using those systems as a starting point, the following characteristics for both the blue and red force main battle tanks are proposed in Table 2.

Table 2. Main Battle Tank Baseline Characteristics

Main Battle Tank Characteristic	Blue Force Baseline Parameter	Red Force Baseline Parameter
Armor (mm)	600	900
Main Weapon Armor Penetration (mm)	120	125
Main Weapon Rounds (#)	40	45
Reloading Rate (rounds/min)	10	9
Effective Range (meters)	3500	3000
Secondary Weapon Armor Penetration (mm)	30	30
Secondary Weapon Rounds (#)	900	2000
Speed (mph)	45	44
Protection System Angle	300	N/A
Protection System Reaction Time	70	N/A
Protection System Reload Rate (mins)	1.5	N/A

Note that the characteristics of the red force main battle tank will remain constant within the simulation model while the characteristics of the blue force main battle tank will be varied to assess the impact of alternative design decisions on operational performance.

MANA-V Terrain Implementation

An attractive feature of MANA-V as an analysis program to assess the effectiveness for DoD systems is the ability to modify terrain features to implement variability in operational environments. Terrain within MANA-V is defined by three values: going (which impacts ease of movement), cover (which impacts probability of hit), and concealment (which impacts probability of detection). Each value is scaled from 0 to 1 and acts as a multiplier on each associated probabilistic event (movement speed, probability of hit, and probability of detection). Based on the operational concept descriptions presented in Figure 8 and Figure 9, a terrain map is generated. That terrain map, shown in Figure 16, implements the multipliers found in Table 3.

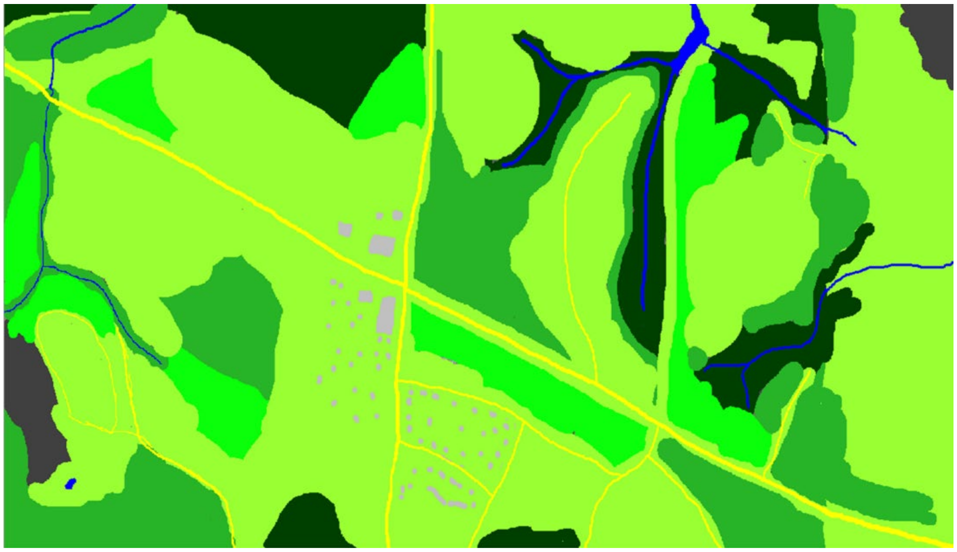


Figure 16. MANA-V Terrain Map (Fernandez et al., 2018)

Table 3. MANA-V Terrain Multipliers

Terrain Feature	MANA-V Color	Going	Cover	Concealment
Forest	Dark Green	0.40	0.20	0.90
Flat Rural	Yellow-Green	0.80	0.10	0.60
Light Brush	Bright Green	0.70	0.20	0.70
Dense Brush	Olive Green	0.60	0.30	0.80
Roads	Yellow	0.90	0.10	0.10
Water	Blue	0.10	0.10	0.10

A full description of the operational simulation used for this project is available in Fernandez et al. (2018). To take advantage of the batch run capability within MANA-V, an appropriate experimental design approach tailored to execution for simulation models is developed.

Experimental Design

Based on the baseline system characteristics presented in Table 2, a range of potential values for each design characteristic of the NATO main battle tank are developed. The objective of using ranges rather than fixed values is to support investigation of alternative system configurations, each of which employs the same operational activities and system functions described in the system architecture. Table 4, from Fernandez et al. (2018), presents the maximum and minimum values for each main battle tank design characteristic.

Table 4. Main Battle Tank Characteristic Ranges (Fernandez et al., 2018)

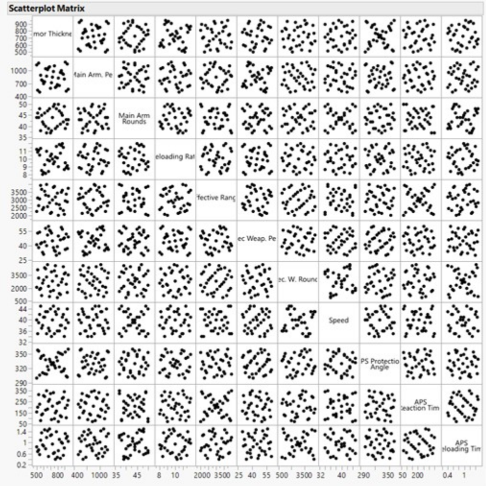
MBT Characteristic	Minimum	Maximum
Armor (mm)	500	950
Main Armament Armour Penetration (mm)	500	1200
Main Arm Rounds (#)	37	50
Reloading Rate (rounds/min)	8	12
Effective Range (meters)	2000	4000
Secondary Weapon Armour Penetration	30	60
Secondary Weapon Rounds (#)	900	4750
Speed (MPH)	34	45
APS Protection Angle	300	360
APS Reaction Time	70	350
APS Reloading Rate	0.4	1.5

Prior to an in-depth discussion of the experimental design, each of the characteristics are described in more detail. The intent is to provide clarity regarding the implementation of each characteristic in MANA-V.

- Armor Thickness: The armor thickness of the main battle tank (defined in the Tangibles tab of MANA-V in millimeters)
- Main Armament Armor Penetration: The depth of armor that the primary weapon can penetrate (defined in the Weapons tab of MANA-V in millimeters)
- Main Armament Number of Rounds: The number of rounds available for use with the primary weapon system (defined in the Weapons tab of MANA-V as a simple count)
- Main Armament Reloading Rate: The time to reload the primary weapon system (defined in the Weapons tab of MANA-V in seconds)
- Effective Range: The maximum effective range of the primary weapon system on the main battle tank (defined in the Weapons tab of MANA-V in meters)
- Secondary Armament Armor Penetration: The depth of armor that the secondary weapon can penetrate (defined in the Weapons tab of MANA-V in millimeters)
- Secondary Armament Number of Rounds: The number of rounds available for use with the secondary weapon system (defined in the Weapons tab of MANA-V as a simple count)
- Speed: The maximum movement speed for the main battle tank (defined in the Tangibles tab of MANA-V in miles per hour)

- Armor Protection System (APS) Protection Angle: The angle at which the external APS can detect and classify incoming rounds (the APS is defined in MANA-V as a separate weapon that operates in conjunction with the main battle tank, accordingly the protection angle is defined in the Weapons tab of MANA-V as an integer with an allowable range of 0–360)
- APS Reaction Time: The delay for the APS to engage incoming rounds (this is defined in the Weapons tab in MANA-V in shots per minute)
- APS Reloading Rate: The time required to wait between cycles of the APS (defined in MANA-V in minutes)

To support investigation of each variable, a space filling design tailored for use with simulation experiments is selected. Specifically, a nearly orthogonal Latin hypercube (NOLH) design developed by Sanchez (2011) is used to define the system design configurations examined in MANA-V. The NOLH design approach defines 33 system configurations for assessment in MANA-V. To assess the appropriateness of those 33 system configurations, Figure 17 presents a scatterplot and correlation matrix. Note that the scatterplot matrix provides a visual characterization of the 11-dimensional design space (shown as 2-dimensional projections of the space).



Correlations

	Armor Thickness	Main Arm. Pen	Main Arm Rounds	Reloading Rate	Effective Range	Sec Weap. Pen	Sec. W. Rounds	Speed	APS Protection Angle	APS Reaction Time	APS Reloading Time
Armor Thickness	1.0000	0.0007	0.0402	0.0002	-0.0198	0.0195	0.0181	-0.0249	0.0165	-0.0053	0.0144
Main Arm. Pen	0.0007	1.0000	0.0128	-0.0001	0.0109	0.0033	-0.0024	-0.0020	-0.0139	-0.0014	0.0173
Main Arm Rounds	0.0402	0.0128	1.0000	0.0162	-0.0373	0.0048	-0.0016	-0.0272	0.0333	-0.0187	0.0067
Reloading Rate	0.0002	-0.0001	0.0162	1.0000	0.0013	-0.0118	-0.0073	-0.0057	0.0120	-0.0083	-0.0001
Effective Range	-0.0198	0.0109	-0.0373	0.0013	1.0000	0.0025	-0.0013	-0.0152	0.0148	0.0093	0.0214
Sec Weap. Pen	0.0195	0.0033	0.0048	-0.0118	0.0025	1.0000	0.0176	-0.0230	-0.0012	-0.0196	0.0052
Sec. W. Rounds	0.0181	-0.0024	-0.0016	-0.0073	-0.0013	0.0176	1.0000	-0.0056	0.0104	0.0102	0.0112
Speed	-0.0249	-0.0020	-0.0272	-0.0057	-0.0152	-0.0230	-0.0056	1.0000	-0.0326	-0.0256	-0.0256
APS Protection Angle	0.0165	-0.0139	0.0333	0.0120	0.0148	-0.0012	0.0104	-0.0326	1.0000	-0.0183	-0.0320
APS Reaction Time	-0.0053	-0.0014	-0.0187	-0.0083	0.0093	-0.0196	0.0102	-0.0256	-0.0183	1.0000	0.0082
APS Reloading Time	0.0144	0.0173	0.0067	-0.0001	0.0214	0.0052	0.0112	-0.0256	-0.0320	0.0082	1.0000

The correlations are estimated by Row-wise method.

Figure 17. Scatterplot and Correlation Matrix for CF-MBSE Employed NOLH Experimental Design

Each two-dimensional projection shows a complete coverage of the full space. This indicates that the design is likely appropriate and will provide a complete coverage of the design space. The correlation matrix provides a quantitative confirmation of this suspicion. The maximum absolute pairwise correlation between any two design variables is 0.0402, indicating that there is near-zero correlation between each of the input variables and ensuring that the assumptions inherent in any subsequent regression modeling approach will not be violated. Accordingly, the design is accepted and will be utilized to define the system configurations run in the MANA-V model. To account for the stochastic nature of MANA-V, each system configuration will be replicated 30 times, defining a total of 990 simulation runs.

As a note, the employment of the APS was examined in additional detail. Because the APS is an augmentation of the existing capabilities of the main battle tank beyond a straightforward improvement to a design characteristic, the presence of the APS is treated as a binary on/off variable for both the offensive and defensive scenario. This defines four different MANA-V models for assessment. The first model includes the APS as a design characteristic utilized in the defensive scenario. The second model removes the APS as a design characteristic and simulates the defensive scenario. The third model includes the APS as a design characteristic utilized in the offensive scenario. The fourth model removes the APS as a design characteristic and simulates the offensive scenario. The full complement of 990 design points was run for each scenario, resulting in a total of 3,960 model runs.

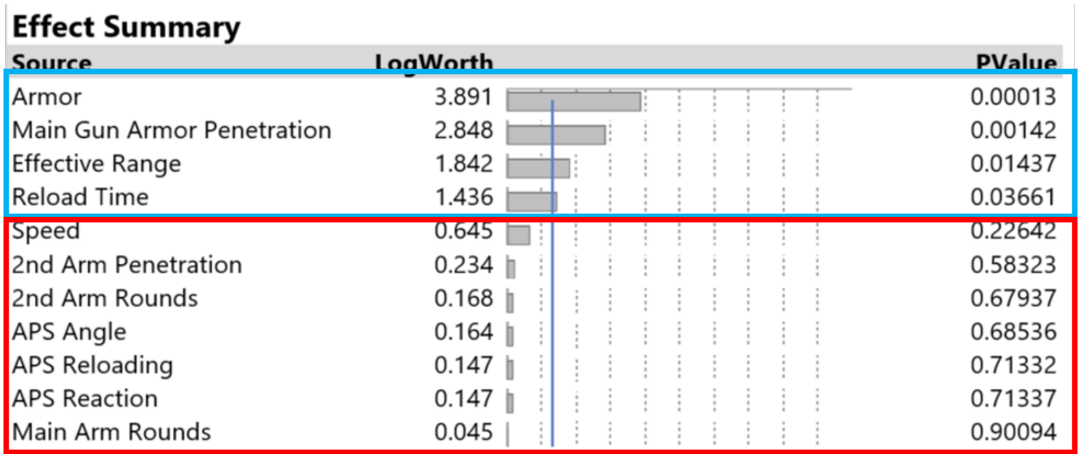
Analysis and Assessment

The analysis of the operational models is focused on each of the MOEs defined in the Capability Requirement step of the CF-MBSE approach. Recall that there were three MOEs associated with both the defensive and offensive scenarios (Deny/Obtain Access, Survive Attack, and Destroy Enemy Forces). Given that both the survive attack and destroy enemy forces MOEs are subsidiary to the overall goal of denying or obtaining access, the Deny/Obtain

Access MOEs will be the focus of the analysis in this demonstration. This also simplified the execution of the MANA-V simulation model by reducing the total number of model termination conditions. For the defensive model, the simulation terminates after total destruction of either force or when the red force captures a high value target (representing control of the checkpoint). Similarly, for the offensive model, the simulation terminates after total destruction of either force or when the blue force captures a high value target (representing control of the checkpoint).

Defensive Scenario Assessment

Recall that the defensive scenario simulated a blue tank platoon defending a NATO checkpoint from an attack by a red force equipped with four main battle tanks, seven ATGMs, and 30 support infantry. Regression analysis was conducted to identify the system characteristics that had the largest impact on the MOE of interest, the probability that the blue force successfully denied access to the checkpoint. For the defensive scenario where the APS was included, the regression results indicate that the presence of the APS dominates the results. Figure 18 presents the regression results, conducted in a statistical software package called JMP and presented in Fernandez et al. (2018), for the defensive scenario where the APS is employed (note that "Armor" as a variable corresponds to the binary presence of the APS).

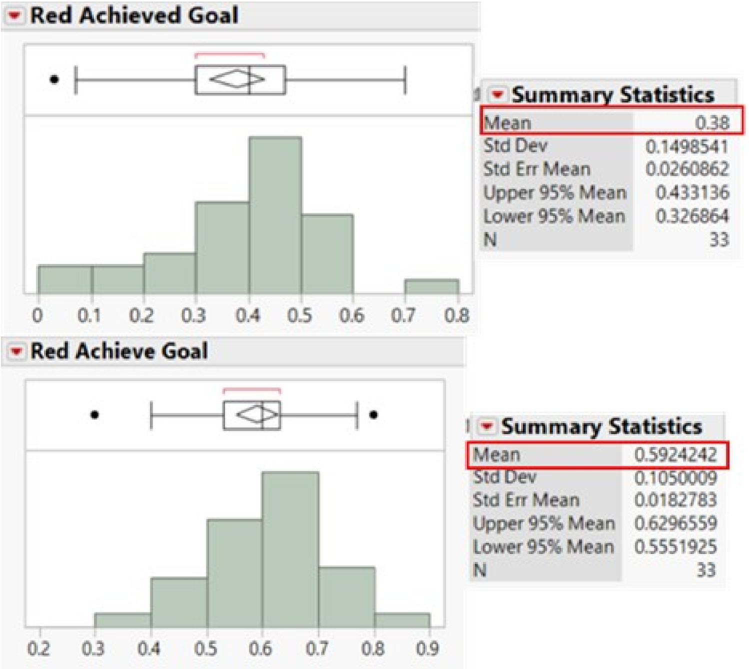


Effect Summary

Source	LogWorth	PValue
Armor	3.891	0.00013
Main Gun Armor Penetration	2.848	0.00142
Effective Range	1.842	0.01437
Reload Time	1.436	0.03661
Speed	0.645	0.22642
2nd Arm Penetration	0.234	0.58323
2nd Arm Rounds	0.168	0.67937
APS Angle	0.164	0.68536
APS Reloading	0.147	0.71332
APS Reaction	0.147	0.71337
Main Arm Rounds	0.045	0.90094

Figure 18. Regression Results: Defensive Scenario with APS (Fernandez et al., 2018)

The regression results indicate that the presence of the APS has the largest impact on model variability. Interestingly, the specific characteristics of the APS (angle, reloading time, and reaction time) are not identified as statistically significant. This suggests that, in a defensive scenario, the presence of the APS improves the ability of the blue force to deny access, but detailed design of individual characteristics of the APS have substantially less impact. The results are perhaps more pronounced when segmenting the data and calculating a simple average of the probability that the red force reached its goal (the MANA-V terminating condition associated with a failure to Deny Access) for cases where the APS is equipped vs. absent. The top of Figure 19 presents the average performance when the APS is equipped, while the bottom presents the average performance when the APS is absent.



Summary Statistics

Mean	0.38
Std Dev	0.1498541
Std Err Mean	0.0260862
Upper 95% Mean	0.433136
Lower 95% Mean	0.326864
N	33

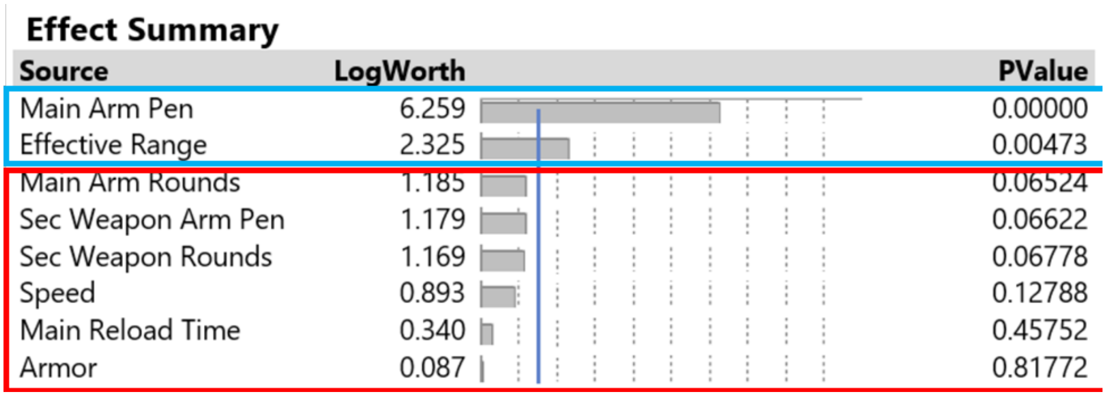
Summary Statistics

Mean	0.5924242
Std Dev	0.1050009
Std Err Mean	0.0182783
Upper 95% Mean	0.6296559
Lower 95% Mean	0.5551925
N	33

Figure 19. Average Probability of Success for Deny Access MOE by APS Presence (Fernandez et al., 2018)

Note that the average probability that the red force is able to take control of the checkpoint increases from 38% to 59% when the APS system is removed. This is consistent with the interpretation of the regression results from Figure 18, which indicate that the presence of the APS has a larger impact on model variability than any other main battle tank design characteristic.

While the potential impact of the APS is apparent from both the regression analysis and the assessment of the mean performance, the regression analysis also suggests that three other main battle tank design characteristics are statistically significant. From Figure 18, these characteristics are the Main Gun Armor Penetration, the Main Gun Effective Range, and the Main Gun Reload Time. To examine the generalizability of these insights, regression analysis is performed for the defensive scenario without the APS employed. Figure 20 presents the results of that analysis.



Effect Summary

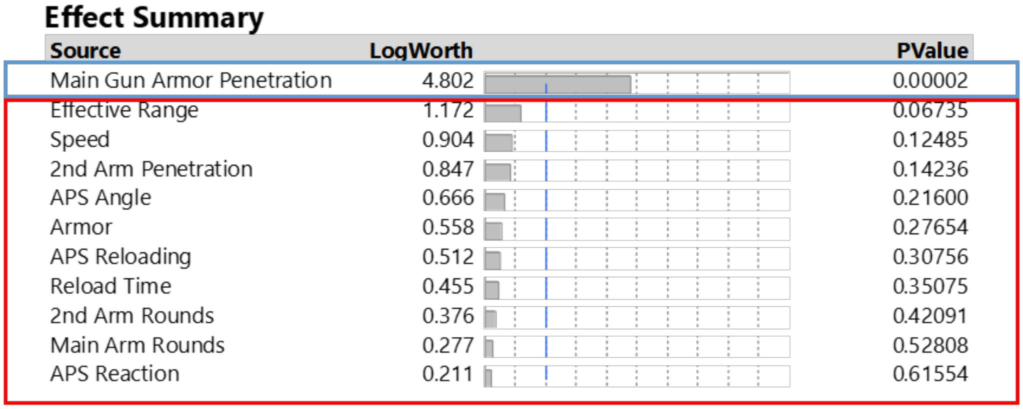
Source	LogWorth	PValue
Main Arm Pen	6.259	0.00000
Effective Range	2.325	0.00473
Main Arm Rounds	1.185	0.06524
Sec Weapon Arm Pen	1.179	0.06622
Sec Weapon Rounds	1.169	0.06778
Speed	0.893	0.12788
Main Reload Time	0.340	0.45752
Armor	0.087	0.81772

Figure 20. Regression Results: Defensive Scenario Without APS (Fernandez et al., 2018)

Similar to the results from the defensive scenario when the APS is present, both the Main Gun Armor Penetration and the Main Gun Effective Range are identified as statistically significant. Interestingly, the Main Gun Reload Time is not identified as statistically significant (p-value of 0.45). This suggests that, regardless of the presence of the APS, if defensive scenarios are the primary focus for system design, initial investment should focus on the Main Gun Armor Penetration and the Main Gun Effective Range. To explore the consistency of that recommendation in an alternative scenario, a similar assessment is conducted for the operational scenario.

Offensive Scenario Assessment

Recall that the offensive scenario utilized a larger, battalion sized unit supported by 12 main battle tanks. The red force is assumed to be the same size as in the defensive scenario to represent their ability to rapidly reinforce the checkpoint to mitigate any losses experienced in their attack. As with the defensive scenario, the offensive scenario is run for 990 design points both with and without the APS. Figure 21 presents the regression results for the offensive scenario with employment of the APS.



Effect Summary

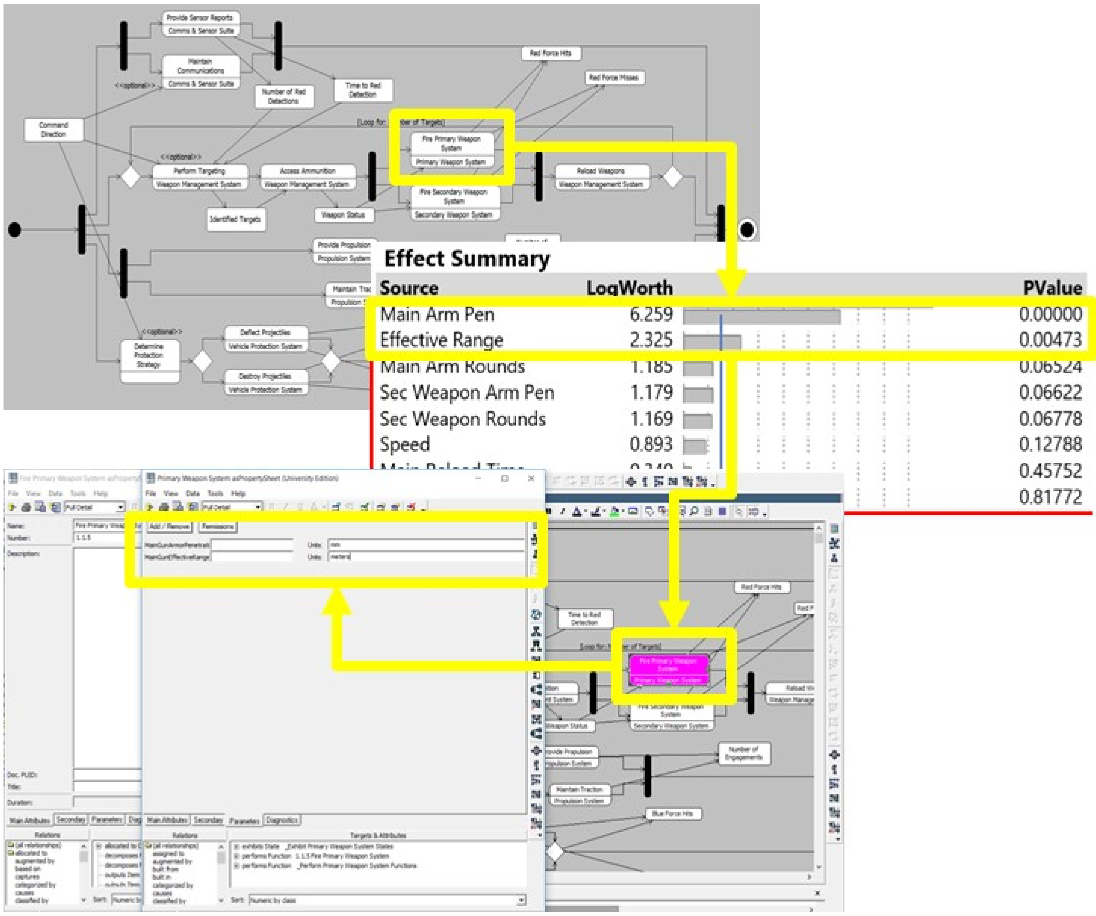
Source	LogWorth	PValue
Main Gun Armor Penetration	4.802	0.00002
Effective Range	1.172	0.06735
Speed	0.904	0.12485
2nd Arm Penetration	0.847	0.14236
APS Angle	0.666	0.21600
Armor	0.558	0.27654
APS Reloading	0.512	0.30756
Reload Time	0.455	0.35075
2nd Arm Rounds	0.376	0.42091
Main Arm Rounds	0.277	0.52808
APS Reaction	0.211	0.61554

Figure 21. Regression Results: Offensive Scenario with APS (Fernandez et al., 2018)

Similar to the defensive scenario, the Main Gun Armor Penetration is identified as statistically significant. However, unlike the defensive scenario, the presence of the APS is not statistically significant (p-value 0.27). This may be attributed to either the increased number of blue force tanks (12 in the offensive scenario vs. four in the defensive scenario) or the impact of a change in the engagement strategy for either the blue or red force. Also, while the effective range is not technically statistically significant at the traditionally applied p-value of 0.05, it does have the largest impact on operational effectiveness for any design characteristics outside of the Main Gun Armor Penetration (p-value 0.067). This suggests that, as with the defensive scenario, these two variables are likely the most impactful design characteristics when assessing the capabilities of the NATO main battle tank.

Assessment Summary

The CF-MBSE approach concludes with an assessment of the system design characteristics that have the largest impact on the MOEs of interest to the system design. In this specific case, it is apparent that, across multiple operational scenarios, the Main Gun Armor Penetration and Main Gun Effective Range are the most appropriate candidates for investment. This can be linked back to subsequent iterations of the CF-MBSE approach. As an example, Figure 22 presents a visualization of an iteration through the products generated in multiple steps of the CF-MBSE approach.



Effect Summary

Source	LogWorth	PValue
Main Arm Pen	6.259	0.00000
Effective Range	2.325	0.00473
Main Arm Rounds	1.185	0.06524
Sec Weapon Arm Pen	1.179	0.06622
Sec Weapon Rounds	1.169	0.06778
Speed	0.893	0.12788
Main Reload Time	0.340	0.45752
		0.81772

Figure 22. CF-MBSE Notional Iteration

On the top of Figure 22 is the DoDAF SV-4 generated in step 2 of the CF-MBSE approach. That SV-4 defined “Fire Primary Weapon System” as a system function, allocated to the Primary Weapon System. The analysis results (highlighted in the center of Figure 22) suggest that this system function, while previously unconstrained, requires additional specification. Accordingly, the SV-4 generated in step 2 of the CF-MBSE should be revised to introduce a parameter associated with the Primary Weapon System that specifies an appropriate value for both the Main Gun Armor Penetration and the Main Gun Effective Range, the two design characteristics that dominated the operational effectiveness analysis.

The intent of this recommendation is not to drive a design decision for the NATO main battle tank, but rather it is to emphasize that the CF-MBSE, through utilization of a computer based MBSE tool, facilitates rapid reuse and update that can add additional detail to subsequent versions of operational analysis.

THIS PAGE INTENTIONALLY LEFT BLANK

Summary and Future Work

The intent of this project is to develop and demonstrate a CF-MBSE framework that takes advantage of recent advances in MBSE to better support system engineering and acquisition. Specifically, it focuses on a process for generation of MBSE-compliant systems architecture representations designed to enable operational modeling and simulation in such a way that the results could be easily integrated into assessments of alternative fleet designs and concepts.

This paper presents a literature review of recent efforts in both MBSE and operational modeling. That review may serve as a foundation for follow-on efforts focused on integration or employment of either domain. Additionally, this project proposes and describes a CF-MBSE approach that defines a five-step process to support system design and analysis using MBSE architectural representations, operational simulation models, and system assessment techniques. Critically, the approach is agnostic to the specific modeling language utilized for the development of the systems architecture products. This is done to facilitate applicability to both the broader engineering and more specific DoD acquisition community. The CF-MBSE framework describes the intended utility of each step of the process and provides specific direction for the creation of both SysML and DoDAF compliant products within each step. This allows a user to approach capability assessment from either the traditional engineering perspective where requirements serve as the starting point for design, which emphasizes the decoupling and deconflicting of system behavior and system structure. This also allows a user to approach capability assessment from a DoD acquisition perspective, where high-level capabilities are translated to lower-level operations which specify individual system functions and component structure. Both approaches are detailed and described within the CF-MBSE, and the appropriate architectural products to support both perspectives are identified. Given that the presentation of the CF-MBSE framework is necessarily conceptual in some locations, this project concludes with an application of the CF-MBSE approach to orient design decisions for a conceptual NATO main battle tank around operational capability. The demonstration walks through the generation of initial capability requirements, consistent with both DoDAF standards and specific SysML diagrams. The demonstration subsequently builds a linkage from those capability requirements to architectural representations of both the operational and system perspectives. These perspectives serve as the starting point for the development of an operational simulation model, built in the agent-based program MANA-V. The MANA-V model is described and organized, and an appropriate experimentation approach is presented. Finally, an assessment of the design characteristics of the main battle tank that have the largest impact on operational capability is conducted, and a recommended approach for iteration of the CF-MBSE approach is presented. This walkthrough is intended to ensure that, while the generic representation of the CF-MBSE approach may be conceptual in places, actionable instruction is possible via mirroring of the steps in the demonstration.

While the CF-MBSE framework has potential applicability to structure thinking and guide system engineering and acquisition to a capability-focused approach rather than system-focused approach, there are areas where the work can be expanded and improved. Most notably, the connections between the MBSE models developed in steps 1 and 2 of the CF-MBSE approach and the operational models developed in step 3 of the CF-MBSE can be formalized and automated. Currently, there is extensive research focused on the linkage of structural SysML models (using Block Definition, Internal Block, and Parametric Diagrams) to link directly to supporting simulation models. There is comparatively less research done linking behavioral SysML models to supporting simulation models. This is likely a result of both the technical maturity of the programs that can be linked to SysML diagrams, as well as the freedom associated with the definition of correctness for a behavioral model. Technically, the implementation of a mapping program that translates the design space as defined in SysML structural diagrams is likely less complex than development of a mapping program that provides similar utility for behavioral diagrams. Given that the end state of a structural model is a consistent design that is physically feasible, the end state of a behavior model is comparatively less restricted as a result of the fact that behavioral feasibility is comparatively less constrained. This is not necessarily a problem; rather it is an opportunity to invest in the formalization of behavioral models and development of standards that provide a foundation for the automation of mapping between models.

THIS PAGE LEFT INTENTIONALLY BLANK

List of References

Acheson, P., Dagli, C., & Kilicay-Ergin, N. (2013). Model based systems engineering for system of systems using agent-based modeling. *Procedia Computer Science*, *16*, 11–19.

Baldwin, W. C., Sauser, B., & Cloutier, R. (2015). Simulation approaches for system of systems: Events-based versus agent based modeling. *Procedia Computer Science*, *44*(1), 363–372.

Behdani, B. (2012, December). Evaluation of paradigms for modeling supply chains as complex socio-technical systems. In *Proceedings of the 2012 Winter Simulation Conference (WSC)*, 1–15. IEEE.

Bleakley, G., Lapping, A., & Whitfield, A. (2011, June). Determining the right solution using SysML and model based systems engineering (MBSE) for trade studies. In *INCOSE International Symposium*, *21*(1), 783–795.

Buede, D. M. (2009). *The engineering design of systems: Models and methods*. John Wiley & Sons.

Delligatti, L. (2014). *SysML distilled: A brief guide to the systems modeling language*. Addison-Wesley.

Department of the Army. (2012). *Army doctrine reference publication 3–90: Offense and defense*.

Department of the Army, Training and Doctrine Command, G-2. (2016). *Ground systems: Worldwide equipment guide volume 1*.

Department of Defense. (2018a). *Department of Defense systems engineering strategy*. https://www.acq.osd.mil/se/docs/2018-DES.pdf

Department of Defense. (2018b). *Summary of the 2018 National Defense Strategy of the United States of America*.

Department of Defense Chief Information Officer. (2015). *DoDAF: DoD architecture framework version 2.02*. http://dodcio.defense.gov/Library/DoDArchitectureFramework.aspx

Dickerson, C. E., & Mavris, D. (2013). A brief history of models and model based systems engineering and the case for relational orientation. *IEEE Systems Journal*, *7*(4), 581–592.

Estefan, J. A. (2008). Survey of model-based systems engineering (MBSE) methodologies. *INCOSE MBSE Focus Group*, *25*(8), 1–12.

Fernandez, A., Hoose, J., Paz, N., Sirico, A., & Zuniga, J. (2018). *A systems engineering analysis of NATO main battle tank capabilities to 2030*. [Capstone report]. Naval Postgraduate School.

Frey, T. M., & Valencia, M. C. (2010). Enhancing systems engineering agility with modeling and simulation. *John Hopkins APL Technical Digest*, *29*(2), 186–196.

Friedenthal, S., Griego, R., & Sampson, M. (2007, June). INCOSE model based systems engineering (MBSE) initiative. In *INCOSE 2007 Symposium*, 11.

Friedenthal, S., Moore, A., & Steiner, R. (2009). *A practical guide to SysML: The systems modeling language*. Morgan Kaufmann.

German, E. S., & Rhodes, D. H. (2017). Model-centric decision-making: Exploring decision-maker trust and perception of models. In *Disciplinary Convergence in Systems Engineering Research* (pp. 813–827). Springer, Cham.

Gold, R. (2016, October). Mission engineering. In *19th Annual NDIA Systems Engineering Conference*.

Hart, L. E. (2015, July). Introduction to model-based system engineering (MBSE) and SysML. In *Delaware Valley INCOSE Chapter Meeting*. Mount Laurel, NJ.

Headquarters Department of the Army. (1996). *FM 17-15: Tank platoon.*

Huldt, T., & Stenius, I. (2019). State of practice survey of model-based systems engineering. *Systems Engineering*, *22*(2), 134–145.

International Council on Systems Engineering. (2015). Systems engineering handbook: A guide for system life cycle processes and activities (Version 4.0). *INCOSE International Symposium*, *25*(1), 678–686.

Kapos, G. D., Dalakas, V., Nikolaidou, M., & Anagnostopoulos, D. (2014). An integrated framework for automated simulation of SysML models using DEVS. *Simulation*, *90*(6), 717–744.

LaSorda, M., Borky, J. M., & Sega, R. M. (2018). Model-based architecture and programmatic optimization for satellite system of systems architectures. *Systems Engineering*, *21*(4), 372–387.

Law, A. (2014). *Simulation modeling and analysis* (5th ed.). McGraw Hill.

Law, A. M. (2009, December). How to build valid and credible simulation models. In *2009 Winter Simulation Conference*, 39–47. IEEE.

Maheshwari, A., Kenley, C. R., & DeLaurentis, D. A. (2015, October). Creating executable agent based models using SysML. In *INCOSE International Symposium*, *25*(1), 1263–1277.

MacCalman, A., Kwak, H., McDonald, M., & Upton, S. (2015). Capturing experimental design insights in support of the model-based system engineering approach. *Procedia Computer Science*, *44*, 315–324.

Object Management Group. (2006, July). OMG systems modeling language (OMG SysML) tutorial. In *INCOSE International Symposium*, *9*, 65–67.

Prosnik, G. (2010). Materials from “Systems 101: Fundamentals of systems engineering planning, research, development, and engineering.” In J. Snoderly & B. Zimmerman (Eds.), *DAU distance learning program.* Defense Acquisition University (DAU)/U.S. Department of Defense (DoD).

Reid, J. B., & Rhodes, D. H. (2016, March). Digital system models: An investigation of the non-technical challenges and research needs. In *Conference on Systems Engineering Research*.

Richardson, J. (2016). *A design for maintaining maritime superiority*. Department of the Navy. https://www.navy.mil/cno/docs/cno_stg.pdf

Richardson, J. (2018). *A design for maintaining maritime superiority 2.0*. Department of the Navy. https://www.sdmac.org/media/uploads/a_design_for_maintaining_maritime_superiority_ver_2.0..pdf

Russell, M. (2012). Using MBSE to enhance system design decision making. *Procedia Computer Science*, *8*, 188–193.

Ryan, M., Cook, S., & Scott, W. R. (2013). *Application of MBSE to requirements engineering—Research challenges.*

Sanchez, S. (2011). *NOLHdesigns spreadsheet.* http://harvest.nps.edu/

Sanchez, S. M., & Wan, H. (2012). Work smarter, not harder: A tutorial on designing and conducting simulation experiments. In *Proceedings of the 2012 Winter Simulation Conference (WSC)*.

Santner, T., Williams, B., & Notz, W. (2003). *The design and analysis of computer experiments*. Springer Series in Statistics.

Siebers, P. O., Macal, C. M., Garnett, J., Buxton, D., & Pidd, M. (2010). Discrete-event simulation is dead, long live agent-based simulation! *Journal of Simulation*, *4*(3), 204–210.

Weilkiens, T. (2008). *Systems engineering with SysML/UML: Modeling, analysis, design*. Morgan Kaufmann Publishers.

Zimmerman, P., & Dahmann, J. (2018, October). *Digital engineering to support mission engineering.* Presentation at the 21st Annual NDIA Systems and Mission Engineering Conference.

Zimmerman, P. (2017). *DoD digital engineering strategy.* Presentation at the 20th Annual NDIA Systems Engineering Conference.

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

www.acquisitionresearch.net