33rd Annual INCOSE international symposium
hybrid event
Honolulu, HI, USA
July 15 - 20, 2023

Using the Unified Architecture Framework in Support of Mission Engineering Activities

James N Martin
The Aerospace Corporation
14745 Lee Road, Chantilly, VA, USA
James.N.Martin@aero.org

Kyle E Alvarez
The Aerospace Corporation
7250 Getting Heights, Col Springs, CO, USA
Kyle.E.Alvarez@aero.org

Abstract. Mission Engineering is the deliberate planning, analyzing, organizing, and integrating of current and emerging operational and system capabilities to achieve desired warfighting mission effects. A Mission Engineering process was developed to capture the approach defined in the DOD's Mission Engineering Guide. The purpose of this is to help train new mission engineers and merge this with standard approaches for capturing mission architectures. The Unified Architecture Framework (UAF) provides a framework of standardized views from which to model different aspects of an architecture, including the various concepts and properties of the mission being engineered. The Mission Engineering steps are tied to the workflow steps in the Enterprise Architecture Guide for UAF to help inform mission engineers of which UAF views can be used during the Mission Engineering effort. This paper will discuss mapping between the UAF workflow and steps in the Mission Engineering Guide and how to use UAF when doing Mission Engineering activities.

Table of Contents

1 Introduction

The size and complexity of national defense operations present unique challenges to architects and systems engineers who must determine the best mission architectures that can reliably achieve desired strategic, tactical, and operational effects in an uncertain and often hostile environment. At its core, this work revolves around connecting mission activities with required resources (i.e., the systems, software, technologies and people for the fielded capabilities) and identifying areas where this connection is insufficient or does not exist (resulting in a variety of capability gaps). Architectural models can be used to illuminate alternative ways and means to possibly close the capability gaps while living within the constraints of funding, doctrine, policy, organization, etc.

Traditional systems engineering evaluates the performance of individual systems against a set of requirements or a concept of operations [ISO 15288 2015], whereas mission architects evaluate how a collection of systems (which can be composed as “Mission Engineering Threads”) will perform a variety of relatively broad missions. The complexity and scope of this enterprise requires a methodical approach for architects from definition of the problem to characterization of a suitable mission reference architecture. Mission Engineering can provide such an approach for conducting architecture assessments of this kind. UAF can facilitate modeling of the architecture in a standardized manner to increase the speed and effectiveness when building those models.

Mission Engineering. Mission Engineering can be used when developing reference architectures for military force operations. These mission reference architectures can help guide force design efforts when they examine various ways and means for achieving desired operational and strategic effects. A variety of documents are created throughout this process. Documentation captures the challenges and drivers, evaluation of mission operations and mission threads, assumptions and constraints underpinning the analysis, and the results of the analysis. The Mission Engineering process defined in the Mission Engineering Guide [DOD 2020] can be conducted in either a paper-based environment or a digital environment. This paper serves to illustrate a digital approach using the UAF modeling workflow [OMG 2022b].

Digital Engineering. In a paper-based environment, documents pertaining to the architecture are typically created using a variety of software applications and sent to stakeholders for their review. Whenever a document is changed, members of the study team will need to manually update other documents affected by the changes. Conducting Mission Engineering in a purely document-based environment is not ideal since maintaining traceability and version control becomes more challenging over time. Time is wasted attempting to trace documents to each other and ensuring they are always up to date. In a digital environment, documentation exists as models held in digital form in Model Based Systems Engineering (MBSE) tools. Changes made to the MBSE models are instantly made available to all stakeholders of interest. Linkages between models in the digital tools ensure that full traceability is maintained throughout.

Using UAF to Define the Mission Architecture. There is significant interest amongst several DOD organizations to move from a paper-based environment to a digital environment. The goal of this paper is to show how the Unified Architecture Framework (UAF), an enterprise modeling standard for defining architectures [OMG 2022a], can be used to digitally model the results of Mission Engineering activities by mapping steps in the Mission Engineering process to steps in the UAF Enterprise Architecture Workflow defined in the EA Guide for UAF [OMG 2022b].

2 Background

2.1 Mission Engineering Guide

Mission Engineering, as defined by the 2020 DOD Mission Engineering Guide, is the deliberate planning, analyzing, organizing, and integrating of current and emerging operational and system capabilities to achieve desired warfighting mission effects [DOD 2020]. Mission Engineering is used by defense organizations to better understand the military force's future capability gaps by conducting analysis on mission threads that cover scenarios the force could face. It is expected that by discovering these capability gaps, those who conceive of new mission designs can then work towards closing those gaps.

To help the DOD conduct Mission Engineering, a guide was developed to document the recommended Mission Engineering approach. The guide starts from defining a problem facing the force, converting it into operational mission threads, defining mission engineering threads that implement the mission threads, running analysis on the mission thread and mission architectures, and documenting the conclusions. The process covers how to conduct the analysis side of Mission Engineering but does not provide sufficient detail on the descriptive modeling needed to support the analysis. Descriptive modeling covers the creation of mission threads that describe how to carry out the mission and the mission architectures that carry out the mission. UAF can be used to model the descriptive aspects of Mission Engineering in a standard and robust manner that is more effective and efficient, thereby saving time and effort when capturing the mission architecture.

2.2 UAF Standard

The UAF specification consists of four main components as illustrated in Figure 1 [OMG 2022a]. View specifications are organized in a two-dimensional grid, and these provide direction to tool vendors and to those who are creating the architecture views regarding what types of model elements are pertinent to those views. These view specifications are instantiated in architecture modeling tools that conform to the UAF specification and drive creation of the relevant architecture views. The views in UAF are an evolution of the views provided in DODAF [DOD 2009].

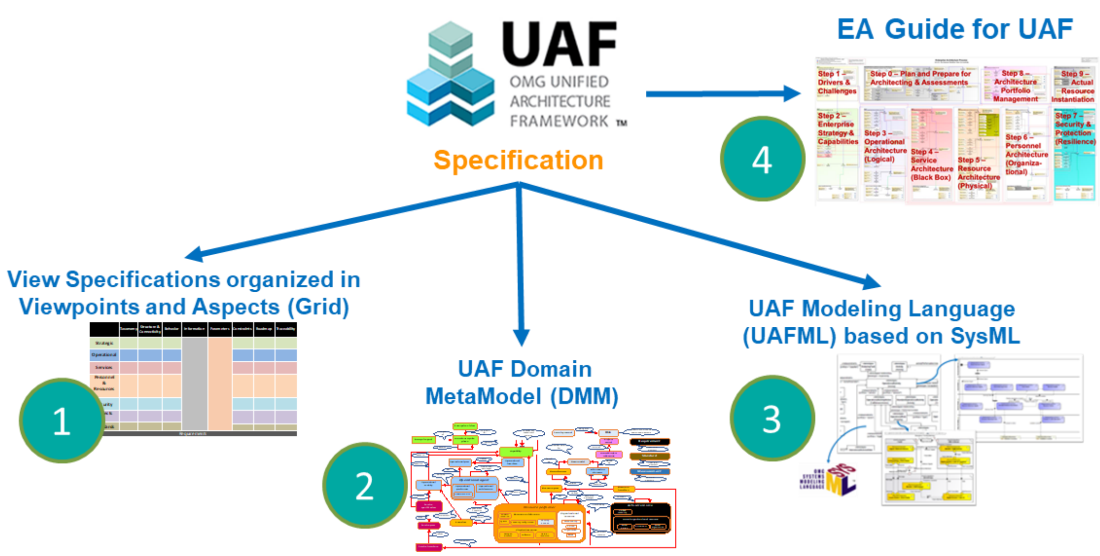


Figure 1. Major Components of the Unified Architecture Framework

The Domain Metamodel (DMM) in UAF establishes the underlying foundational modeling constructs to be used in modeling an enterprise and major entities within the enterprise. The UAF Modeling Language (UAFML) specifies how SysML modeling constructs can be used to create the views defined by the view specifications. The Enterprise Architecture (EA) Guide [OMG 2022b, Martin 2021] provides a structured way to create the views defined in UAFML and is intended to be used in conjunction with the model views provided in the UAF Sample Problem for a Search and Rescue enterprise [OMG 2022c]. The UAF Specification documents (including a Sample Problem) can be downloaded from the OMG webpage: www.omg.org/spec/UAF/About-UAF. Further information on UAF can be found at https://www.omgwiki.org/uaf.

2.3 View Specifications in UAF

The UAF Grid (Figure 2) is comprised of rows that represent typical stakeholder domains (called *viewpoints* in UAF) that can be used when modeling an enterprise architecture. The Grid has columns that represent the architecture *aspects* (in UAF v1.1 these were called 'model kinds') that correspond to "part of an entity's character or nature" [ISO 2022]. This Grid is a structuring formalism for organizing the 82 view specifications defined within the UAF standard.

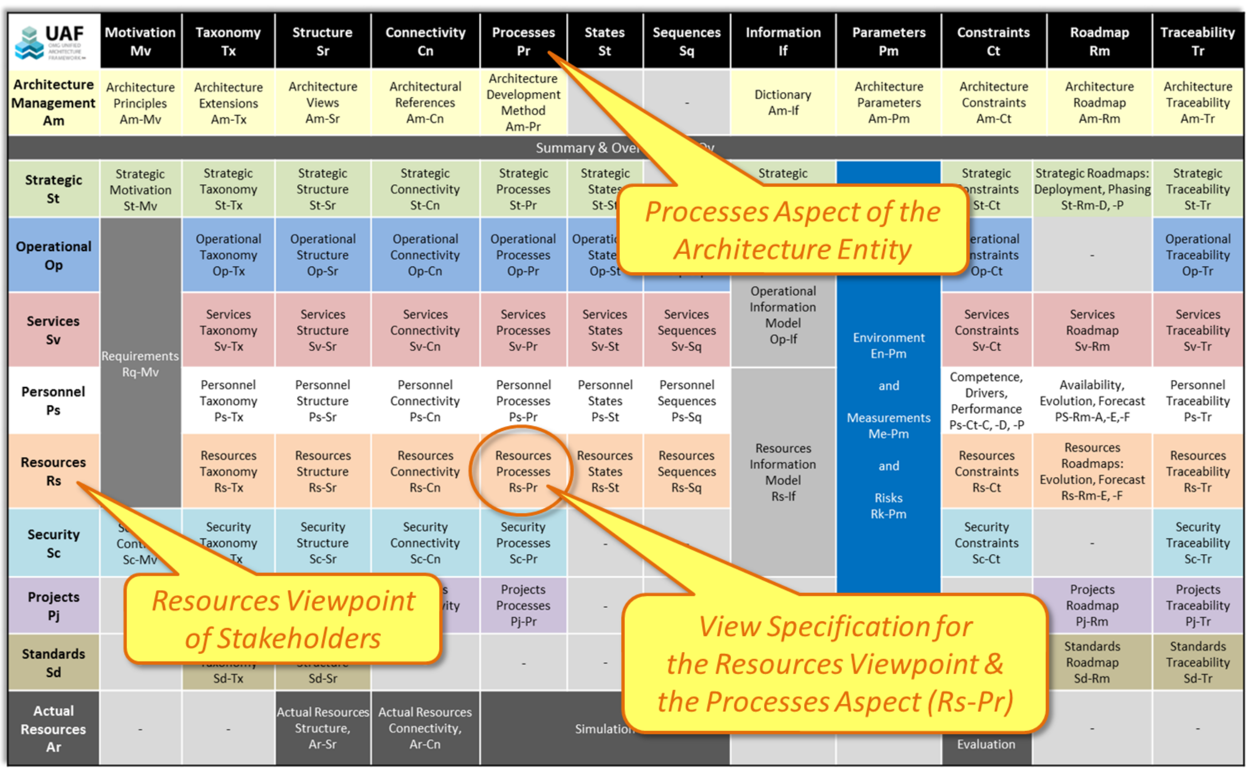


Figure 2. View Specifications Organized by a Two-dimensional Grid in UAF

2.4 EA Guide for UAF

The EA Guide for UAF covers architecting of the enterprise as well as architecting (at a high level) of a major entity within the enterprise. This architecture description workflow in the Guide (shown in Figures 3 and 4) can be used in conjunction with processes for conceptualization and evaluation of architecture. The underlying UAF workflow is consistent with the Architecture Elaboration process in the 42020 standard [ISO 2019] and has the following intended uses:

— Process reference model for Enterprise Architecture (EA) Process Guide to be included in the OMG standard for the Unified Architecture Framework (UAF),

— Reference model as the basis for an EA Modeling Methodology that defines associated methods, patterns, templates, tools and techniques for each process step,

— Process framework for project planning and architecture definition activities, and

— Training and certification on architecture frameworks and modeling approaches.

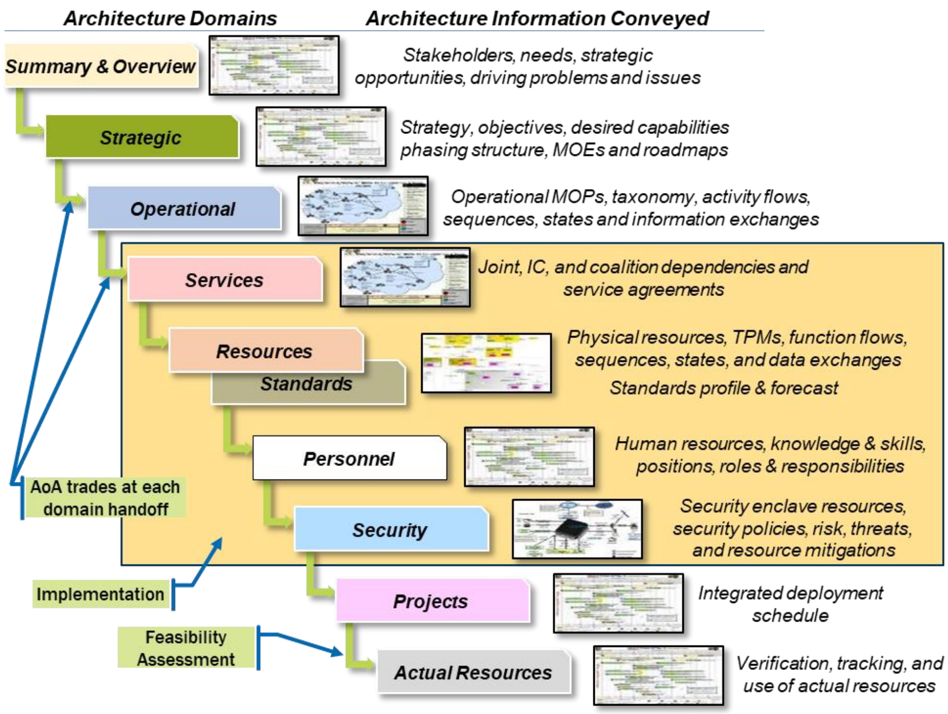


Figure 3. Major Components in the EA Guide Workflow

Architecture Domains in UAF. The general workflow to implement these architecting activities is illustrated in Figure 3. Each step in the process conveys the architecture information to iteratively produce a definition of the problem space along with a definition of the solution space (i.e., implementation and instantiation). Tradeoffs are identified along the way and architectural decisions are captured in the architecture views as they are fleshed out. There will be some repetition back and forth between the steps to ensure a complete and coherent depiction of the architecture as it unfolds. It is not necessary that it must be implemented in a top-down fashion.

EA Guide Workflow Steps. The workflow in the EA Guide defines "what" to do when creating the UAF views but does not identify or define methods (the "how") or tools needed for each step (since the particular tools used are dependent on the methodology to be employed). The nine steps in Figure 4 follow the basic flow illustrated earlier and these steps are decomposed to the third level to get to the point where individual UAF views are generated for each of the sub-steps.

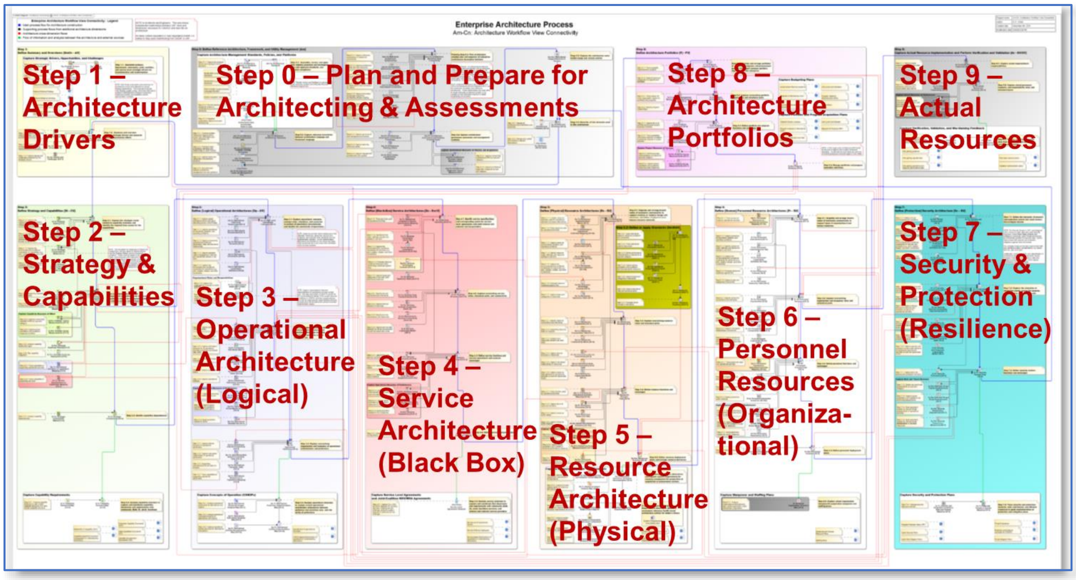


Figure 4. EA Guide Workflow Steps (High Level)

To illustrate this, here are the workflow steps for defining Operational Concepts in the architecture:

Step 3: Define [Logical] Operational Architectures	UAF Architecture Views
Step 3.1: Capture operational concepts - including concept roles, situa-tions, and scenarios in context of operational environments and identify the constraints of operations	Op-Tx: Operational Taxonomy: High Level Operational Concepts [OV-1]
Step 3.1.1: Capture simple operational sketches with users describing all key CONOPS ideas	Op-Tx: Operational Taxonomy: Concept of Operations Sketch [OV-1]
Step 3.1.2: Capture operational environments, regions, theaters, and op-erating conditions	En-Pm: Environment: Operational [N/A]
Step 3.1.3: Capture overarching operational architecture performers, roles, and structural relationships	Op-Sr: Operational Structure [OV-2]
Step 3.1.4: Capture operational rules of engagement, methods, and oper-ational policies in rule form	Op-Ct: Operational Constraints [OV-6a]
Step 3.1.5: Capture the environment and conditional constraints for oper-ations (e.g., operational areas, planning scenarios, threats, locations, etc.)	Op-Ct: Operational Constraints: Defini-tion [OV-6a]
Step 3.1.6: Capture the organizations involved in overall CONOPS	Ps-Tx: Personnel Taxonomy: Organiza-tional Context [OV-4]
Step 3.1.7: Capture the responsibilities of the organizations involved in the CONOPS relative to their roles	Ps-Sr: Personnel Structure: Organiza-tional Responsibilities [OV-4]

The full complement of workflow steps down to the second level is shown in Figure 5. It is tempting to think of this workflow as a process. However, in practice many of the steps are conducted simultaneously by a variety of people involved in the overall architecture development effort. This can be thought of as an architecture modeling and definition "work breakdown structure." The complete set of workflow steps down to the third level of detail are defined in the "EA Guide for UAF" that can be found here: https://www.omg.org/cgi-bin/doc?formal/22-07-10.pdf.

The problem framing approach [Martin 2019] should be used to help identify the most useful UAF views to develop for a particular modeling effort. This approach helps avoid the problem of "model creep" which is why this is one of the activities to be accomplished as part of UAF Step 0.

0. **Define Reference Architecture, Framework, and Architecture Enablers**
"The purpose of this step is to provide information pertinent to the entire architecture and to acquire or develop key enablers to facilitate development and maintenance of the architecture models and views."
0.1. Assemble Standards and Practices
0.2. Conduct Problem Framing
0.3. Plan Architecture Description Standup
0.4. Capture and Monitor Architecture Governance Plan
0.5. Capture Profile and Environment Usage
0.6. Capture Enterprise Terms and Definitions

1. **Define Architecture Drivers and Challenges**
"The purpose of this step is to identify those things that drive the enterprise to do what it does and the associated challenges that present difficulties in addressing these drivers."
1.1. Assemble Strategic Drivers
1.2. Capture Enterprise Challenges and Opportunities
1.3. Organize Architectural Descriptions
1.4. Analyze Strategic Tradeoffs and Decisions

2. **Define Strategy and Capabilities**
"The purpose of this step is to describe the capability taxonomy, composition of capabilities, dependencies between capabilities, and evolution of the capabilities."
2.1. Capture The Strategic Vision
2.2. Capture Capabilities
2.3. Identify Capability Dependencies
2.4. Analyze Capability Relationships

3. **Define [Logical] Operational Architectures**
"The purpose of this step is to describe the requirements, operational behavior, structure, and exchanges required to support (i.e., exhibit) capabilities."
3.1. Capture Operational Concepts
3.2. Capture Operational Behaviors
3.3. Capture Operational Taxonomy
3.4. Analyze Operational Structure

4. **Define Services Architectures**
"The purpose of this step is to define services and to specify required and provided service levels for the services needed to exhibit capabilities and to support operational activities."
4.1. Identify Service Opportunities
4.2. Capture Service Structures
4.3. Define Service Functions
4.4. Define Service Deployment Plans
4.5. Analyze Service Obligations

5. **Define [Implementation] Resource Architectures**
"The purpose of this step is to capture a solution architecture consisting of various resources, such as software, artifacts, capability configurations and natural resources that implement the operational elements and requirements in the operational architecture."
5.1. Establish Resource Taxonomy
5.2. Define Standards Profile
5.3. Capture Resource Structure
5.4. Define Resource Functional Behavior
5.5. Define Resource Deployment Plans
5.6. Capture Resource Requirements and Actual Resources

6. **Define [Human] Personnel Architectures**
"The purpose of this step is to clarify the role of Human Factors when creating architectures in order to facilitate both Human Factors Integration and Systems Engineering."
6.1. Establish Personnel Taxonomy
6.2. Capture Personnel Structure
6.3. Define Personnel Functional Behavior
6.4. Define Personnel Resource Deployment Plans
6.5. Capture Human Resource Requirements

7. **Define [Protection] Security Architecture**
"The purpose of this step is to illustrate security assets, security constraints, security controls, security control families and the measures required to address specific security concerns."
7.1. Establish Security Taxonomy
7.2. Capture Security Structure
7.3. Define Security Behavior
7.4. Analyze Security Plans and Capture Requirements

8. **Manage Project Portfolios**
"The purpose of this step is to describe projects and project milestones, how those projects deliver resources that lead to capabilities, the organizations contributing to the projects and dependencies between projects."
8.1. Establish Project Taxonomy
8.2. Capture Project Structure
8.3. Define Project Activity Behavior
8.4. Manage Project Execution Activities

9. **Capture Actual Resource Instantiation and Support Architecture Evaluation**
"The purpose of this step is to illustrate the expected or achieved actual resource configurations and actual relationships between them."
9.1. Capture Actual Personnel Structure
9.2. Actual Resources Mapping
9.3. Perform Parametric Evaluations

Figure 5. EA Guide Workflow Structure (to Second Level)

3 Mapping Mission Engineering to UAF

UAF provides a standard set of architecture views for describing various aspects of an enterprise and major entities in the enterprise. UAF can readily be used for defining a mission architecture. The six steps of the Mission Engineering approach are shown in Figure 6. As we illustrate in this paper, the UAF workflow can be used to capture the results of each step of the Mission Engineering effort. This section discusses the mapping between Mission Engineering and UAF.

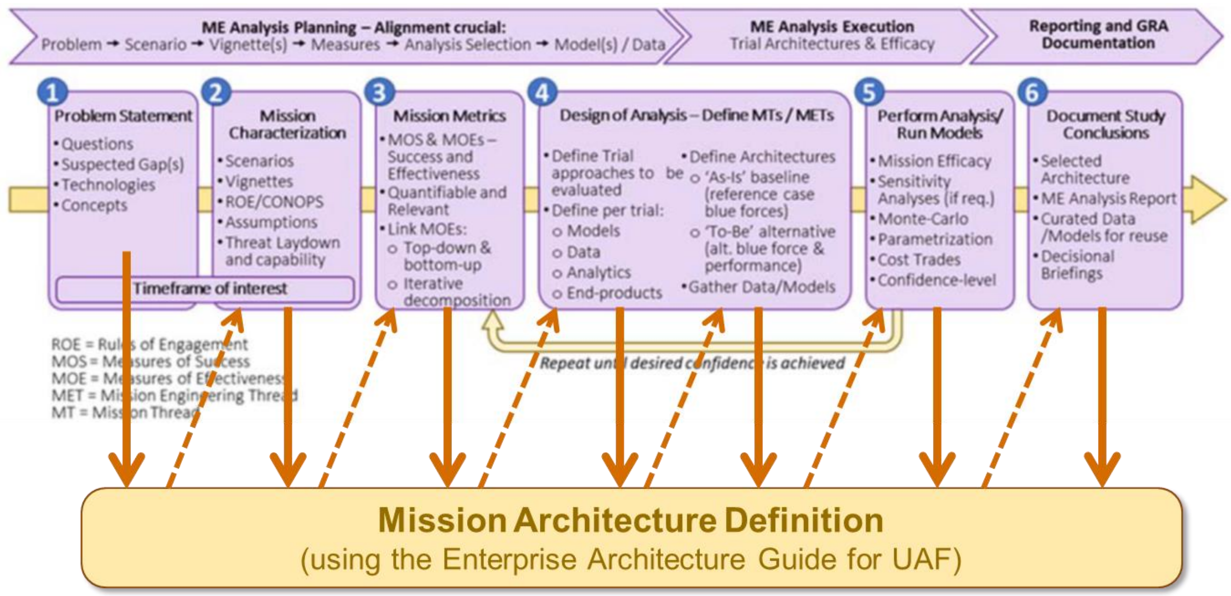


Figure 6. UAF Modeling in Support of the Mission Engineering Effort

3.1 Modeling Using UAF

The models most often used in support of Mission Engineering are typically in SysML format. So, some might ask why not just continue using SysML? What is the advantage of using UAFML?

The UAF Modeling Language[1] (UAFML) is an implementation of the DMM that specifies how the UAF views can be modeled using SysML notation and semantics. Even though the UAFML is based on SysML, there are some significant differences that should be noted. SysML is great for doing the following activities: (a) modeling systems and for doing systems engineering, (b) defining and tracing between levels of abstraction within a system, (c) defining the logical and physical attributes for a system and the mapping of requirements and functions to these attributes [OMG 2019]. The UAF Modeling Language provides all this, plus more:

[1] In version 1.2 of the UAF specification, the UAF Profile (UAFP) was renamed as the UAF Modeling Language (UAFML) to better reflect its intended purpose. Where SysML is a general-purpose language for doing systems engineering, the UAFML is a general-purpose language for modeling an enterprise in support of Enterprise Systems Engineering (ESE) and Enterprise Architecture activities. Of course, UAFML can also be used to model systems, subsystems, major assemblies, products, software applications, but there is usually a transition point where SysML is used primarily at some lower level.

a) **Capability and Enterprise Concepts:** defines the "why" and "what" and "when" before defining the "how"

b) **Services Concepts:** definition of enterprise services (producing and consuming) and traceability to capabilities, operations and implementing resources

c) **Human Factors:** How people and systems interact, along with their expected knowledge & skills (competencies)

d) **Security:** Identifying risk, its mitigation, and integrating security into the architecture

e) **Standards:** definition of and compliance with standards in the architecture

f) **Project Deliveries:** phased milestone approach to capability deployment

g) **System Configuration Over Time:** deployment of systems and their resources and changes in associated system roadmaps and timelines

h) **Tie-in to Non-System Elements in the Architecture:** Easy way to link the entire Architecture to Requirements

i) **Built-in Traceability Between Multiple Views:** Between Layers and Across Layers

Furthermore, the Mission Engineering models if created in UAFML can more readily be flowed down to System Models created using UAF at the services and component organizational levels. In general, UAF is designed to support Enterprise Systems Engineering [Rebovich and White 2010] and, as this paper illustrates, can readily support Mission Engineering activities.

3.2 High Level Mapping

Figure 7 shows a high-level mapping of Mission Engineering steps to UAF workflow steps. For the most part, there is a linear correlation of ME steps to UAF steps. When defining the problem, one can follow the first two UAF workflow steps to cover defining what the architecture will cover and identifying the drivers and challenges. As the mission itself becomes characterized, step two in the UAF workflow is applicable as it covers defining the strategy and capabilities associated with the architecture.

High level mission metrics (MOSs and MOEs) are defined in the UAF Operational Architecture covered in the third step in the UAF workflow. (Each MOS defines how mission success will be measured while each MOE defines how well the desired effects are to be achieved.) Measure of Performance (MOPs) are defined in the UAF Resource Architecture which is covered in step five of the UAF workflow.

In terms of defining the mission architecture and mission threads Steps 3-7 in the UAF workflow can be followed. Each subsequent step provides finer details for supporting the analysis. UAF step eight can capture the project that will implement the reference architecture while the final UAF workflow step can capture the architectures that will feed into analytical models and capture progress on implementing the mission reference architecture.

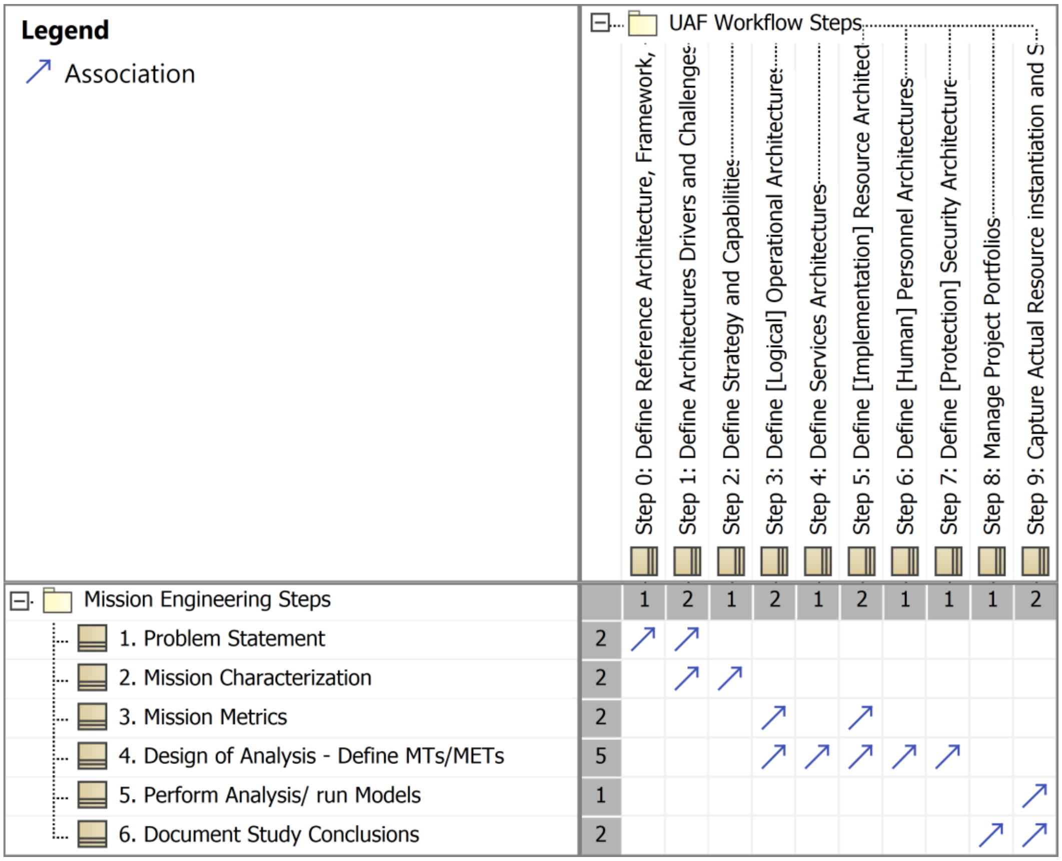


Figure 7. High-Level Mapping from Mission Engineering to UAF Workflow Steps

3.3 Mission Engineering Steps

Figure 8 shows an outline of the Mission Engineering Process contained in the Mission Engineering Guide [DOD 2020]. The first step in the ME process covers the problem statement. The organization initiating the study needs to identify what is the purpose of the study. By identifying a purpose, the questions to be answered by the study can be determined. Finally, the technology areas and mission areas are identified, which will help identify the relevant stakeholders who should participate, or be informed by, the study.

The second step in the Mission Engineering process characterizes the mission. To define the mission itself, operational purpose documents, such as the National Defense Strategy, need to be linked in order to define a time frame, scenarios and vignettes. Defining the operational environment will affect how the mission can be conducted as constraints and threats will change based on the theater of operations. Finally, documenting assumptions behind the study earlier on will pay dividends in terms of conducting a properly comprehensive study.

The third step in the Mission Engineering process covers mission metrics. Measures of Success (MOSs) are used to measure how well the mission is accomplished. Measures of Effectiveness (MOEs) measure how well the tasks that comprise the mission are accomplished. Finally, Measures of Performance (MOPs) measure how well the systems that conduct the mission will perform.

1. **Problem Statement**
   1.1. Articulate Purpose of Study
   1.2. Develop Questions of Interest to be Answered
   1.3. Articulate Mission or Technology Area of Concern
2. **Mission Characterization**
   2.1. Define the Mission
      2.1.1. Link commander's intent with Operational Purpose Documents
      2.1.2. Define the Time Frame
      2.1.3. Define Scenarios
      2.1.4. Define Vignettes
   2.2. Define the Operational Environment
      2.2.1. Link to Defense Planning Scenarios
      2.2.2. Define Geographic Area
      2.2.3. Define the Conflict
      2.2.4. Define the Threat Laydown
      2.2.5. Define Red and Blue Forces
      2.2.6. Define the Order of Battle
      2.2.7. Define Rules of Engagement
   2.3. Define Operational Assumptions and Constraints
      2.3.1. Define Environmental Constraints
      2.3.2. Define Resource Constraints
      2.3.3. Define Force Constraints
      2.3.4. Define Technical Constraints
      2.3.5. Define Technology Roadmaps
3. **Mission Metrics**
   3.1. Define Measures of Success (MOSs)
   3.2. Define Measures of Effectiveness (MOEs)
      3.2.1. Derive MOEs from the Mission Statement and Problem Statement
      3.2.2. Derive MOEs from the Constituent Approaches and Systems Proposed
   3.3. Define Measures of Performance (MOPs)
   3.4. Establish Metric Traceability
      3.4.1. Trace MOEs to MOSs
      3.4.2. Trace MOPs to MOEs
4. **Design of Analysis – Define MTs/METs**
   4.1. Define Mission Architectures
      4.1.1. Define the "As-Is" Architecture
      4.1.2. Define "To-Be" Architectures
   4.2. Define Mission Approach
      4.2.1. Define Mission Threads (As-Is and To-Be)
      4.2.2. Define ME Threads (As-Is and To-Be)
   4.3. Define and Gather Supporting Analytical Models, Data, and Analytics
      4.3.1. Model Development
      4.3.2. Model Management
5. **Perform Analysis/ Run Models**
   5.1. Identify Appropriate Analysis
      5.1.1. Identify Sensitivity Analysis to be performed
      5.1.2. Address if optimization and/or parameterization needs to be performed
      5.1.3. Determine most applicable analytical methods
      5.1.4. Identify and understand error and uncertainty propagation across the system of models
   5.2. Conduct Analysis
      5.2.1. Determine Confidence Levels
      5.2.2. Compute Metrics
      5.2.3. Answer Questions
      5.2.4. Identify Capability Gaps
6. **Document Study Conclusions**
   6.1. Prepare Analysis Report/ Give Decision Briefings
      6.1.1. Discuss the Problem
      6.1.2. Define the Study
      6.1.3. Address issues or Uncertainties
      6.1.4. Describe Conclusions
      6.1.5. Make Recommendations for Further Studies and for Leadership Actions
   6.2. Identify a Reference Architecture
   6.3. Curate Data Models and Architectures

Figure 8. Summary of Mission Engineering Process Steps

The fourth step defines the "As-Is" and "To-Be" mission architectures that pertain to carrying out the mission. The operational behaviors for the mission are defined by "As-Is" and "To-Be" Mission Threads (MTs). Capturing the systems and other resources needed to implement the MT's operational activities will define the Mission Engineering Threads (METs). Analytical models need to be found, developed and managed. These analytical models are used to evaluate performance of the MET assets against the mission metrics.

The fifth step in the Mission Engineering process deals with performing the analysis. The practitioner will need to determine what is the appropriate analysis to conduct and identify sources of errors across the collection of models. After analysis is conducted, confidence levels will need to be assessed to determine if the analysis is usable. If it is, using the computed metrics from the analysis, the questions defined at the start can be answered and capability gaps are identified.

The final step in the Mission Engineering process deals with documenting the study. Analyst reports and decision briefings are developed to recap the study, state the findings, and make recommendations for decision makers. A mission architecture used in the study that was found to be the best performing is identified as a reference architecture. Finally, any models and data created for the study usually will need to be curated for use by other Mission Engineering studies.

3.4 Example Mapping for ME Step 4

Figure 9 shows the mapping between ME Step 4 and UAF step 3. This mapping shows how UAF's Operational Architecture views can be used to model the activities associated with the creation of Mission Architectures, Mission Threads, and analytical modeling efforts. The fourth step in the ME process involves designing the analysis to be conducted by the study. This involves descriptive modeling of the mission architecture and mission threads along with the analytical modeling used for evaluating the performance of the descriptive models. The descriptive modeling involves defining the operational architectures along with the mission threads defining the process and/or operational sequences for carrying out the missions involved.

	UAF Step 3: Define [Logical] Operational Architectures	**Step 3.1: Capture Operational Concepts**	Step 3.1.1: Capture simple operational sketches with users	Step 3.1.2: Capture operational environments, regions,	Step 3.1.3: Capture overarching operational architecture	Step 3.1.4: Capture operational rules of engagement,	Step 3.1.5: Capture the environment and conditional	step 3.1.6: Capture the Organizations Invovled in the Overall	Step 3.1.7: Capture the responsibilities of the organizations	**Step 3.2: Capture operational behaviors**	Step 3.2.1: Capture definition of all CONOPS activities	Step 3.2.2: Capture performer connections and interfaces	Step 3.2.3: Capture information elements for all operational	Step 3.2.4: Specify information exchanges	Step 3.2.5: Capture operational state machines	Step 3.2.6: Capture operational timelines	Step 3.2.7: Capture typical MOPs by type and category	**Step 3.3: Capture operational taxonomy**	Step 3.3.1: Capture internal structure of operational	Step 3.3.2: Capture role-based relationships of operational	Step 3.3.3: Supporting operational performer table for	Step 3.3.4: Trace capabilities to supporting operational	**Step 3.4: Analyze operational structure**	Step 3.4.1: Analyze operational performers for alternatives	Step 3.4.2: Analyze operational role-based impacts for	Step 3.4.3: Define risk assessments by type and category	Step 3.4.4: Capture actual quantitative and qualitative	Step 3.4.5: Build parametric models for MOPs	Step 3.4.6: Capture operational requirements	Step 3.4.7 Capture operational activity implementations to	Step 3.4.8: Capture operational activity structure for
Mission Engineering Step 4: Design of Analysis - Define MTs/METs																															
4.1 - Define Mission Architectures		X	X		X			X	X		X			X	X				X	X											
4.1.1 Define the "As-Is" Architecture																															
4.1.2 Define "To-Be" Architectures																								X	X						
4.2 - Define Mission Approach		X			X	X				X	X					X		X		X			X								X
4.2.1 Define Mission Threads (As-Is and To-Be)																															
4.2.2 Define ME Threads (As-Is and To-Be)												X																			
4.3 - Define and Gather Supporting Analytical Models, Data & Analytics																															
4.3.1 Model Development																												X		X	
4.3.2 Model Management																															

Figure 9. Mapping ME Step 4 Activities to UAF Step 3 Activities

UAF's Operational Architecture is most relevant with the descriptive modeling tasks (mission architectures and mission threads) associated with ME step 4. UAF's operational architecture is used to define solution independent approaches to achieving the desired capabilities of the architecture. For defining Mission Architectures, UAF can model simple operational sketches of the key ideas behind the Concept of Operations (CONOPS) that provides a high-level overview of the mission for stakeholders. This can be done in a UAF Operational Taxonomy (Op-Tx) diagram which is similar to a DODAF OV-1 operational concept diagram.

To provide details about the mission architecture, operational performers can be listed and described in the UAF model. Operational performers are logical elements in the architecture that perform the tasks in the thread without specifying the exact resource used. An Op-Tx diagram can capture a taxonomy of the operational performers and organizations similar to a DODAF OV-2 diagram. The UAF Operational Structure (Op-Sr) diagram can in turn define the internal structure of these operational performers similar to an OV-2 diagram. This is where the strength of UAF shows as different aspects of the architecture can be represented by different diagrams.

For defining mission threads, UAF Operational Processes (Op-Pr) diagrams (similar to a DODAF OV-5a and OV-5b diagrams) can model the CONOPS tasks in a hierarchical tree diagram format and in a process flow diagram. Operational nodes and operational performers and needlines between them are captured in an Operational Structure (Op-Sr) diagram. Information exchanges can be modeled in a UAF Operational Connectivity (Op-Cn) table similar to an OV-3 information exchange matrix. In the Operational Traceability (Op-Tx) table, the operational roles in the mission thread are allocated to resource performers (e.g., systems, people, organizations, capability configurations). This creates a set of logical Mission Engineering Threads (METs).

In terms of analytical modeling, UAF has a few views that can directly support the analysis. An Operational Parametric (Op-Pm) model can be built to evaluate logical MOPs. These parametric models can capture the results of operational and resources analysis and compare different architecture alternatives against one another. Notably, there is no DODAF view that can provide parametric modeling. To keep track of the architecture's implementation of the logical process, a Resources Traceability (Rs-Tr) relation map can be created to show how resource functions trace to operational activities in order to check performer implementation.

3.5 How to Use UAF with Mission Engineering

In general, as the ME process progresses the UAF steps follow suit. UAF workflow step 0 covers the planning and preparing for the architecture modeling effort. UAF step 1 is used to identify the drivers and challenges for the study, based on the strategic and operational driving documents.

The activities associated with defining the mission can be captured in the Strategic viewpoint. This viewpoint captures the enterprise's capabilities the mission engineering study seeks to enhance. UAF step 2 covers strategic architecting. Mission Metrics can be defined in UAF's operational architecture and resource architecture. Generally, MOSs and MOEs can be found in the strategic and operational architectures while MOPs are found in the resource architecture.

As the mission architectures and mission threads become more detailed, additional UAF architecture views can be used. Initially, operational (logical) architectures are modeled to define solution-independent approaches. Eventually the operational elements that map to the desired capabilities are defined. The fourth step in the UAF workflow covers service architecting which is where shared services (e.g., GPS, gov cloud, satellite communications) are identified that are key to mission execution. When it comes to modeling the systems, software, and other non-human solution elements that can implement the logical operations, the UAF Resources viewpoint can provide such modeling. The fifth step in the UAF workflow deals with modeling the resources. If the personnel and organizations operating the resources are defined, the sixth step in the UAF workflow can be followed to model these via the Personnel viewpoint. Finally, if the Mission Engineering study needs to deal with cybersecurity protection or protection from threats, the seventh step in the UAF workflow can be followed to model the security architecture.

The fifth step in the ME process deals with performing analysis of the Mission Threads and Mission Engineering Threads (usually with simulations, but can also be accomplished with wargames, tabletop exercises, role playing in groups, etc.). The Actual Resources viewpoint (covered in UAF step 9) can be used to model real world instances of the architectural elements that will be used in the analytical models.

The sixth step in the ME process covers documenting the conclusions of the ME study. Two UAF viewpoints map to this step. The Projects viewpoint (covered in UAF step 8) can model the portfolio of projects involved with implementing the reference architecture defined by the study. To track implementation of the reference architecture in terms of realizable elements, the Actual Resources viewpoint can be used.

4 Conclusions

In this paper we presented a mapping of the DOD Mission Engineering Process to the UAF workflow and highlighted some of the key considerations when using UAF in this manner. This mapping can assist in descriptive modeling efforts in support of a Mission Engineering project.

> *Note to reviewer: The work described herein is a work in progress. The work will be completed in March and the paper will be updated to include final results for the paper to be delivered in April.*

There is additional work to be accomplished that will help make UAF more compatible with and easier to use during Mission Engineering efforts:

a) Develop a "Mission Engineering Guide for UAF" similar to the "EA Guide for UAF" but with a tailored version containing the most appropriate subset of the UAF workflow steps and views that are most pertinent to Mission Engineering

b) Develop UAF-based modeling patterns and view templates for each step in the Mission Engineering process (that can be used in creating the Sample Model mentioned below)

c) Develop a Sample Model of an example Mission Engineering project to help the modeler understand how to create a model with the right level of details in the most appropriate views for the Mission Engineering steps

d) Develop a training course with exercises in using UAF to create Mission Engineering relevant architecture models and views using the ME-specific modeling patterns

Note to reviewer: There is a new version of the ME Guide coming out December 2022. We will update this paper after the new guide is out to make the work up to date in time for final paper submittal in April 2023.

References

DOD 2009, *DoD Architecture Framework (DODAF)*, Version 2.0, Office of the Deputy Chief Information Officer. Washington, DC: U.S. Department of Defense (DOD). (https://dodcio.defense.gov/library/dod-architecture-framework/)

DOD 2020, *Mission Engineering Guide*, Office of the Deputy Director for Engineering, Office of the Under Secretary of Defense for Research and Engineering. Washington, DC: U.S. Department of Defense (DOD), November 2020. (https://ac.cto.mil/wp-content/uploads/2020/12/MEG-v40_20201130_shm.pdf)

ISO 2015, *Systems and Software Engineering — System Life Cycle Processes,* International Standards Organization (ISO), ISO/IEC/IEEE 15288-2015.

ISO 2019, *Software*, *Systems and Enterprise — Architecture Processes,* International Standards Organization (ISO), ISO/IEC/IEEE 42020-2019.

ISO 2022, *Software*, *Systems and Enterprise — Architecture Description,* International Standards Organization (ISO), ISO/IEC/IEEE 42010-2022.

Martin, James 2019, “Problem Framing: Identifying the Right Models for the Job,” INCOSE International Symposium, July 2019, Orlando, Florida, USA.

——— 2021, “Enterprise Architecture Guide for the Unified Architecture Framework (UAF),” INCOSE International Symposium, July 2021, Virtual Conference.

OMG 2019, *Systems Modeling Language*, Version 1.6, Object Management Group, https://www.omg.org/spec/SysML/About-SysML/.

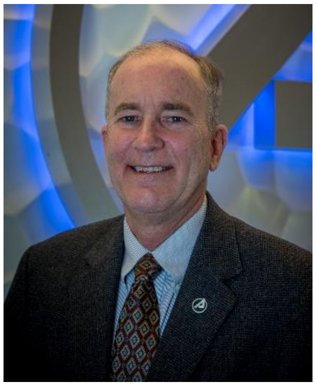
——— 2022a, *Unified Architecture Framework*, Version 1.2, Object Management Group, https://www.omg.org/spec/UAF/About-UAF/.

——— 2022b, *Enterprise Architecture Guide for the Unified Architecture Framework (Informative)*, Version 1.2, Object Management Group.

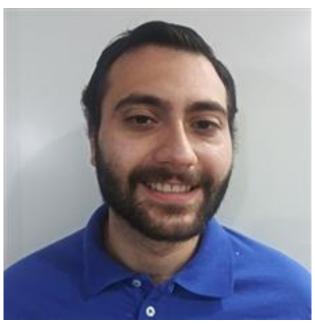
——— 2022c, *Unified Architecture Framework Sample Problem (Informative)*, Version 1.2, Object Management Group.

Rebovich G, and B White (eds.) 2010, Enterprise Systems Engineering: Advances in the Theory and Practice, CRC Press.

Biography



James N Martin is a member of the UAF Revision Task Force with OMG and was lead editor for the ISO 42020 standard on Architecture Processes. He is an Enterprise Architect and a Distinguished Systems Engineer at The Aerospace Corporation developing solutions for information systems and for space domain enterprises. He was a key author on the BKCASE project in development of Enterprise Systems Engineering articles for the SE Body of Knowledge (SEBOK). Dr. Martin led the working group responsible for developing ANSI/EIA 632, a US national standard that defined the processes for engineering a system. He previously worked for Raytheon Systems Company and AT&T Bell Labs on airborne and underwater systems and on mobile telecommunication systems. His book, Systems Engineering Guidebook, was published by CRC Press in 1996. Dr. Martin is an INCOSE Fellow and was leader of the Standards Technical Committee. He was founder and was until recently leader of the Systems Science Working Group. He received from INCOSE the Founders Award for his long and distinguished achievements in the field.



Kyle E Alvarez is a systems engineer with The Aerospace Corporation developing Digital Engineering models in the System of Systems Engineering Office supporting various US Air Force and Space Force government programs. Mr. Alvarez earned a BS and MS in Aeronautical and Astronautical Engineering from Purdue University. He plans on leveraging systems engineering expertise for use on large projects supporting the space enterprise to guarantee mission success in a timely and affordable manner.