A STUDY OF EXECUTABLE MODEL BASED SYSTEMS ENGINEERING FROM DODAF USING SIMULINK

THESIS

Michael H. Ryan, BS
First Lieutenant, USAF

Weston J. Hanoka, BS
First Lieutenant, USAF

AFIT/GSE/ENV/12-S05DL

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States. The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government.

AFIT/GSE/ENV/12-S05DL

A STUDY OF EXECUTABLE MODEL BASED SYSTEMS ENGINEERING FROM DODAF USING SIMULINK

THESIS

Presented to the Faculty

Systems and Engineering Management

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Systems Engineering

Weston J. Hanoka, BS
First Lieutenant, USAF

Michael H. Ryan, BS
First Lieutenant, USAF

September 2012

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT/GSE/ENV/12-S05DL

A STUDY OF EXECUTABLE MODEL BASED SYSTEMS ENGINEERING FROM DODAF USING SIMULINK

Weston J. Hanoka, BS
First Lieutenant, USAF

Michael H. Ryan, BS
First Lieutenant, USAF

Approved:

Brent T. Langhals, Lt Col, USAF (Chairman)	Date
John M. Colombi, Ph.D. (Member)	Date
Michael E. Miller, Ph.D. (Member)	Date

AFIT/GSE/ENV/12-S05DL

Abstract

Diagrams and visuals often cannot adequately capture a complex system's architecture for analysis. The Department of Defense Architectural Framework (DoDAF), written to follow the Unified Modeling Language (UML), is a collection of mandated common architectural products for interoperability among the DoD components. In this study, DoDAF products from as-is Remotely Piloted Aircraft (RPA) Satellite Communication (SATCOM) systems have been utilized for the creation of executable architectures as part of an Executable Model Based Systems Engineering (EMBSE) process. EMBSE was achieved using Simulink, a software tool for modeling, simulating and analyzing dynamic systems.

This study has demonstrated that DoDAF products can be created and executed following the rules of UML for analysis. It has also shown that DoDAF products can be utilized to build analysis models. Furthermore, these analysis models and executable architectures have been presented to a panel of experts on the topic. The comments and study results show a desire for executable architectures as well as their viability as presented in Simulink. This study concludes there is a need, a use and a method to implement objective analysis using EMBSE from DoDAF products in Simulink for current and future DoD systems.

Dedication

We would like to dedicate this thesis to all the deployed US and allied armed forces men and women who are away from their friends and family, risking their lives so that we can continue to live freely.

Acknowledgments

We would like to express our sincere appreciation to our research advisor, Lt Col Brent Langhals, for his support throughout the course of this thesis effort. He has guided us from an idea to a successful end product that we would have otherwise likely not have achieved. He was able to overcome the challenges of advising a team with one member deployed overseas and the other located three time zones away. We would also like to thank members from MILSATCOM that have supported us, in particular: Lt Col Mark Brykowytch and Nuna Bosler, who not only supported us by providing relevant materials for our research, but also gave critical feedback throughout the course of the study from their expertise; Sam Griffin - his models ended up being the basis for some of our research. Lt Col Daniel Harvala - He and his team served as part of our expert panel and gave very beneficial feedback for us that we were able to use for our conclusions. The support we have received truly has helped us exceed our original expectations for this study and we are very grateful.

Weston Hanoka

Michael Ryan

Table of Contents

Page

Abstract iv

Dedication iv

Acknowledgments v

List of Figures viii

List of Tables x

I. Introduction 1

General Issue 1
Problem Statement 3
Research Objectives 4
Research Focus 4
Investigative Questions 5
Methodology 5
Assumptions 6
Summary 6

II. Literature Review 8

Chapter Overview 8
Department of Defense Architectural Framework (DoDAF) 8
DoDAF Shortfalls 11
Benefits of Executable Architectures 13
Deriving Executable Architectures from DoDAF 14
Modeling theory and techniques. 14
Colored Petri Nets. 16
Simulink. 18
Selecting the Tool and Potential Analysis Techniques 19
Remotely Piloted Aircraft Communications Architecture 20
Summary 21

III. Methodology 23

Chapter Overview 23
Approach 23
Executable Architecture for Analysis 24
DoDAF Products. 25
Simulink Modeling from UML. 27

Study Experts 29
Summary 30

IV. Analysis and Results...31

Chapter Overview 31
Results of Executable Modeling: 31
Results from the Questionnaire on Study Experts: 44
Model Specific Feedback. *47*
Overall Feedback. *48*
Summary 49

V. Conclusions and Recommendations 51

Chapter Overview 51
Conclusions of Research by Objective 51
Significance of Research..53
Limitations 54
Recommendations for Action 54
Recommendations for Future Research 55
Summary 55

Appendix A Expert Questionnaire...57

Appendix B RPA DoDAF Viewpoints..61

Appendix C Additional Figures and Tables 69

Appendix D DoDAF Mapping to Simulink...71

Appendix E Screenshots of OV-5b Executable Architecture 75

Appendix F Further Questionnaire Results Analysis..77

Bibliography 79

List of Figures

Page
Figure 1. DoDAF Viewpoints and Descriptions 10

Figure 2. Example CPN of a Simple Protocol 17

Figure 3. Structural Concept Mapping 18

Figure 4. Analysis of DoDAF Products 25

Figure 5. DoDAF Viewpoints used for Model Assessment 1 32

Figure 6. System Latency Model 34

Figure 7. OV-5 DES Model in Simulink 35

Figure 8. Authorizations in Queue 36

Figure 9. Authorizations Submitted 37

Figure 10. Model Assessment 2 39

Figure 11. Monte Carlo GUI 40

Figure 12. Model 2 Output 40

Figure 13. Model Assessment 3 Approach 41

Figure 14. Cost Model GUI 42

Figure 15. Cost Model Output 43

Figure 16. Results by Question 45

Figure 17. DoDAF OV-1: *As-Is* RPA Communications Architecture 61

Figure 18. DoDAF OV-1: *Could-Be* RPA Communications Architecture 62

Figure 19. OV-5b Provide Satellite Access Authorization 63

Figure 20. DoDAF SV-2: System Resources Flow Description 64

Figure 21. DoDAF SV-6: System Resource Flow Matrix 65

Figure 22. DoDAF OV-3: Operational Resource Flow Matrix 66

Figure 23. DoDAF SV-9: Services Technology and Skills Forecast............................... 67

List of Tables

Page
Table 1. Software Platform Selection Criteria 19

Table 2. Analysis Techniques 20

Table 3. DoDAF Views and Descriptions 26

Table 4. Statistical Results from the Questionnaire 46

Table 5. All Results for EMBSE and Analysis in Simulink Questionnaire 47

Table 6. Law and Policy DoDAF Supports 69

Table 7. DoDAF Meta-model Groups to Viewpoints and DoD Key Processes 70

Table 8. Mapping DoDAF Activity Diagram OV-5b to Simulink 71

Table 9. Mapping DoDAF Activity Diagram OV-5b to Simulink Toolbox SimEvents 73

A STUDY OF EXECUTABLE MODEL BASED SYSTEMS ENGINEERING FROM DODAF USING SIMULINK

I. Introduction

General Issue

It is increasingly evident with progressively more complex and interconnected systems of systems and communication technology that there is a need for real time simulation to address deficiencies and areas of improvement which static diagrams fail to capture. Over the years, studies have been accomplished to address such issues with the Department of Defense's (DoD's) ever more complicated systems and how to utilize the mandated Department of Defense Architectural Framework (DoDAF) to create such simulations. A previous study by Beal et al. (2005) applied DoDAF and executable architectures to study graphically distributed Air Operations Centers. AbuSharekh et al. (2007) utilized DoDAF 1.0 series to model executable architectures with temporal relations. Griendling and Marvis (2011) utilized DoDAF compliant executable models to analyze system of system alternatives. In Systems Engineering, we refer to these simulations as executable architectures. There are many definitions for architectures, but one in particular is that a system's architecture is "the fundamental and unifying system structure defined in terms of system elements, interfaces, processes, constraints, and behaviors" (Rechtin, 2009).

DoDAF goes far into detail, and clearly addresses all or most aspects of the definition of a system's architecture. However, the issue lies in that, once complete, DoDAF can often end up as a compilation of documents in which the only method for evaluation of the system in question is subjective reasoning by the individuals overseeing the requirements being met. Integrated architectures are explained to be the foundation for interoperability within the DoD (Mittal, 2006); however, DoDAF doesn't allow the ability to test this interoperability in an objective environment (AbuSharekh, Kansal, Zaidi, & Levis, 2007). Garcia (2007) identifies additional shortfalls, "The DODAF currently does not include Monte Carlo simulation, trade-off analysis, game theory projections or other complexity modeling analytical support tools (Markovian or analytical hierarchical processes support)." DoDAF and the directives that mandate it will be described in more detail in the literary review chapter.

This issue isn't just inherent to DoDAF architectures, but in systems architecting itself. In fact, in the same book that defines the art of systems architecting, there is little to no mention of evaluating the actual architectural framework through executable modeling and simulation. An actual architecture of a building can be tested through modeling for stresses, joints, stability etc., but how does a system's architecture get tested? This can be done in a similar manner, through simulation and modeling theory.

There are many literary works which describe in detail how complicated systems of systems and their behaviors can be simulated and tested for integration, redundancies, efficiencies and other areas of improvement, yet we still today see power points and static diagrams which attempt to address systems so complicated, a single diagram could take up an entire wall. Many of these systems and communications between systems elements and interfaces are beyond the scope of the human mind. In today's integrated Air Force and DoD components, communication pathways are progressively more vulnerable as we come into the battlefield with systems such as the Remotely Piloted Aircraft (RPA) that have to communicate with many entities while performing its duties. DoDAF, in its static form, does not also allow for testing of such communication pathways, timeliness, vulnerabilities, redundancies, bottlenecking or other important command and control (C2) and communication measures. In essence, it has been identified that Executable Model Based Systems Engineering (EMBSE) is required in addition to the DoDAF products to run accurate system threads and simulations for objectively managing requirements, objectives and goals for all stakeholders.

Problem Statement

DoDAF products are a requirement in the acquisitions process, but often are incomplete and presented in UML fashion through PowerPoint, Microsoft Visio, or an architectural tool allowing for static UML documents to be built. There needs to be a method to dynamically analyze architectural products for efficiency, completeness as well as requirements and stakeholder satisfaction. The advanced concepts division of MILSATCOM, which has been tasked with creating and analyzing the as-is communications architecture of current DoD Remotely Piloted Aircraft (RPA) operations, has offered to provide DoDAF products for evaluation and proof of concept EMBSE. Thus, an opportunity exists to discover if DoDAF products can be utilized in executable architecture modeling techniques to yield useful results beyond that of current models. Successful executable models would demonstrate the capabilities of DoDAF in simulation for detailed objective analysis of System of Systems, processes and networks.

Research Objectives

After considering past research and current modeling techniques described in Chapter 2 of this thesis, the following research objectives are proposed:

1. Demonstrate that an executable architecture can be derived from DoDAF views in Simulink. (Note: Simulink is the tool used to create executable models for this research and is further discussed in Chapters 2 and 3)

    A. A successful demonstration will have variable data inputs and produce applicable outputs

    B. The model must be derived from DoDAF compliant viewpoints and documents only. Additional inputs should be annotated and discussed.

2. Evaluate the effectiveness of executable architectures in evaluating DoDAF Models

    A. This objective will determine whether errors, misrepresentations, and gaps in a given DoDAF viewpoint can be identified with a Simulink executable architecture.

    B. Any errors or improvements can then be flowed back to the original system architecture

3. Determine if Simulink is an effective tool for analyzing DoDAF compliant architectures

    A. Answers the question: *Is this a value added method of producing executable architectures for the DoD?*

The answer to these objectives will be an assessment of whether producing executable architectures from DoDAF compliant models is worth the cost, time and other resources required for EMBSE.

Research Focus

The research in this thesis focuses on proof of concept creation of executable architectures built explicitly from DoDAF views, in a common platform capable of EMBSE and dynamic analysis. From the basic proof-of-concept creations, a briefing and a survey will be put together to present to a panel of study experts. The results of the survey and comments received will be used to formulate conclusions on the objectives and suggestions for future EMBSE.

Investigative Questions

Our initial question in this study begins with how the DoDAF products are comprised. Investigation must begin into the relations between the DoDAF products and categorizing them into those which can be executed and those which can be used as supporting material. This then brings us into our next question, what constitutes an executable architecture and what would be the analysis techniques of the executable architecture models? A literary review has been conducted to assist in answering this question. In order to execute an architectural model, there needs to be a software or tool capable of automation and simulation. What software tool or environment is capable of building executable architectures and conducting various analysis techniques? The literary review has compared possible tools and explained how we ultimately selected the software platform, Simulink. Finally, the most important question is what is the value added in utilizing EMBSE for executable architecture and dynamic analysis? To assist in answering this question, study experts from the acquisitions community, familiar with the material and systems, were asked to participate in a briefing and demonstration, and giving their feedback through a common questionnaire.

Methodology

Utilizing past research into creating Executable Architectures from DoDAF views, it will be determined which DoDAF views will be initially required for the as-is executable architecture analysis. Executable architecture analysis techniques will be investigated as well as the various software tools or platforms available for analysis. Initial models will be created based on a foundation from the investigation. Final models will be presented to experts in RPA communications architecture for validity and conclusions. These DoDAF models will be the basis for analysis using executable architecture and other analysis methods.

Assumptions

In order to successfully research and use case studies, several assumptions were made. The first assumption is that members of the expert panel were knowledgeable in MILSATCOM RPA communications architectures and could accurately evaluate products of the case studies. Since the study only had the ability to operate Simulink in the unclassified environment, DoDAF viewpoints used in the research were assumed to be incomplete. This limitation was overcome by internally creating any additional DoDAF viewpoints required that would still prove to work as a proof of concept, without pushing the research into a classified domain.

Summary

In this study, DoDAF products from as-is RPA SATCOM communication systems have been utilized for the creation of executable architectures as part of the Executable Model Based Systems Engineering (EMBSE) process, using Simulink as the software tool and platform for building the models as well as executing and analyzing the architectures. Chapter 2 lays out previous work and research done into DoDAF, Simulink, analysis methods, and executable architectures. Chapter 3 outlines the methodology taken to conduct the study, develop the results and reach conclusions. The results and products of this methodology are covered in Chapter 4. The analysis of the results will be used make conclusions and specific recommendations into next actions and areas for future research, discussed in chapter 5.

II. Literature Review

Chapter Overview

The goal of the Literature Review will be to explore existing research into executable model based systems engineering (EMBSE) and its applications to the Department of Defense Architectural Framework (DoDAF). A number of reports and scientific articles on existing models, executable architectures, and DoDAF mapping into EMBSE were assessed for relevance and potential guidance. There were a few candidate tools for mapping DoDAF into an executable model, so these tools were also reviewed to determine the ideal software to meet the intended goals. Finally, the Remotely Piloted Aircraft (RPA) systems represented by the DoDAF products utilized to create the executable architectures in the case studies will be introduced.

Department of Defense Architectural Framework (DoDAF)

Department of Defense Architectural Framework (DoDAF) provides guidance to allow for joint, multinational and DoD components to have a common architectural framework. This guidance includes the development, representation and understanding of such a framework. A common framework is mandated so that architecture descriptions can be compared, related and reused across organizational boundaries. DoDAF includes structures (often noted as viewpoints or models), rules and high level processes for developing the architectures of systems. DoDAF version 2.0 was signed for approval 28 May 2009 and the current version at the time of this thesis is DoDAF 2.02. There are several federal laws and policies which call for the need of an enterprise architecture to support decision making throughout DoD organizations. A list of these can be found in Appendix C.

DoDAF is composed of eight viewpoints, and each viewpoint is further composed of DoDAF described models or fit-for-purpose views. These can be depicted as graphics, tables or even textual documents. Fit-for-purpose is often described throughout V2.0 to describe an architecture and/or its viewpoints that are customized or focused to meet the needs of the stakeholders, decision makers and process owners. The eight DoDAF viewpoints and a brief description can be seen in the following graphic taken from DoDAF V2.0 section 3.4.2.

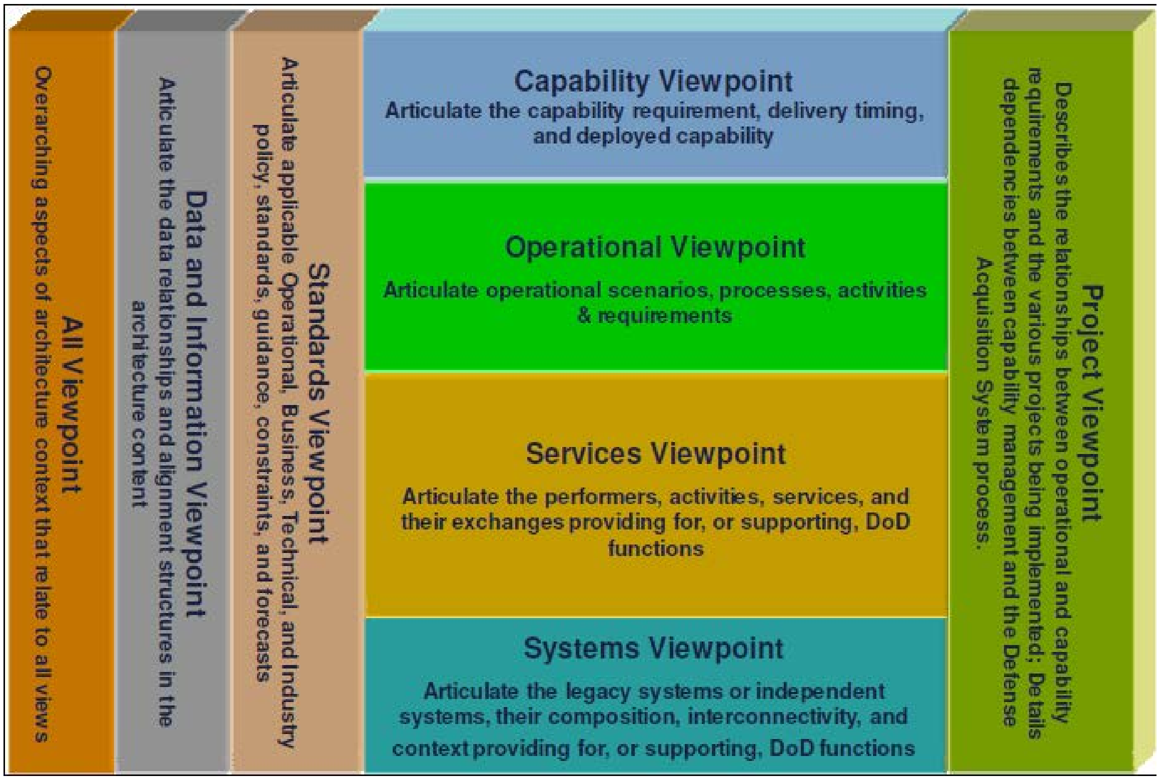


Figure 1. DoDAF Viewpoints and Descriptions

There is also a supporting data model known as the DoDAF Meta Model (DM2) which defines the data structure and architectural relationships or information within in the architecture. A DM2 contains a Conceptual Data Model (CDM), a Logical Data Model (LDM) and a Physical Exchange Specification. Not all of the DoDAF described models have to be created, but there are regulations and instructions from the DoD and the Chairman, Joint Chiefs of Staff (CJCS) that have particular presentation view requirements. For a more in depth description of DoDAF, please refer to *DoDAF V2.02 Web*. A mapping of DM2 to viewpoints and key DoD processes can be seen in Appendix C.

Furthermore, DoDAF V2.0 describes two categories of analytical activity: Static analysis and dynamic analysis. Static analysis is described as the analysis based on data extracted from the architecture descriptions to make a value judgment. Dynamic Analysis is described as the analysis which is "based on running an executable version of the architectural data to observe the overall behavior of the model" (Department of Defense, 2012). It is interesting to note here that DoDAF 2.0 doesn't go much further into detail for executable architectures than this, providing little direction as to how to analyze an architecture to determine the how the stakeholder requirements might have been met, or how to improve on efficiency. Further discussion is in Chapter 3 for specific viewpoints and models being used to aid in the creation of the executable architecture.

DoDAF architectures are often created in platforms that use the Unified Modeling Language (UML) or Systems Modeling Language (SySML) as the common language. These languages are similar and provide a common way to represent data in a system's architecture. As part of the proof-of-concept, a mapping from DoDAF products in SySML/UML to Simulink is attempted and discussed as part of results. The common platforms used in the DoD to create DoDAF products are Sparx Systems' Enterprise Architect, Microsoft Visio and PowerPoint.

DoDAF Shortfalls

There have been several papers in the past which have identified the inability of early forms of DoDAF (versions 1.0 and 1.5) to allow for a systems engineering analysis of products in terms of executable architecture. One of the earliest such papers to address the shortfalls in the DoD's common enterprise architecture in terms of executable architectures was (Levis & Wagenhals, 2000). The latest from Dr. Levis discusses DoDAF's inability to allow the derivation of an executable architecture strictly from DoDAF models. The difficulties often were with initial conditions and temporal issues not addressed therein (AbuSharekh, Kansal, Zaidi, & Levis, 2007). Furthermore varying modeling assumptions not traceable to DoDAF products for an executable model may yield "models with a variety of behavioral properties" (Griendling & Marvis, 2011). This presents an issue when there are multiple organizations involved in a joint project, or even if different stakeholders interpret assumptions differently. Also, early versions did not include specification of scenarios in which time-state transition diagrams could be generated. Because of these inherent issues, executable models could not be made to be algorithmic or automatic in nature when only DoDAF products are used. These architectural models couldn't provide insight into logical, behavioral and performance aspects of systems (Griendling & Marvis, 2011).

DoDAF 2.0 Series has made tremendous progress in specifying many aspects of the system which improved upon previous versions. The key change in the 2.0 series is that DoDAF now focuses on a "data-centric" process, instead of a "product-centric" process. Products as described by the 1.0-1.5 series are now labeled as views and viewpoints for broad conceptual understanding. "The basis of the Architecture Development Process is now the Data Meta-model Groups" (Department of Defense, 2012). A DoDAF Meta-model (DM2), containing a Conceptual Data Model (CDM), a Logical Data Model (LDM), and a Physical Exchange Specification (PES) has been added and created as a part of the new data-centric approach. Fit-for-purpose views and models customized to the system have also added benefit to the executable architectures.

With the use of a DoDAF add-in to SPARX System's Enterprise Architect software as well as other beta software tools in development, there have been great strides toward turning DoDAF architectural models straight to code. While these are significant improvements, DoDAF views and DM2 models when produced are still not executable themselves and produce only static analysis results requiring subjective value judgments. They remain a complicated way to understand the system and its impacts and do not have the benefit of providing insight into performance, logical and behavioral aspects of architecture.

Benefits of Executable Architectures

Executable architecture enables the ability to assess the impacts on System of Systems, which is increasingly important in net-centric systems of the present and future technologies (Griendling & Marvis, 2011). Mission level impacts, integration into a joint environment, system integration and alternatives can all be assessed early in the acquisitions life cycle through an executable architecture analysis. Executable architectures will also differ from simulations, as they are directly derived from the architectural model itself. With a directly derived architecture from DoDAF and an executable architecture tool, the following have been identified as potential benefits: the architecture model itself can be verified for internal self-consistency; operational concepts can be simulated, observed dynamically, verified and refined; operational plans can be examined and assessed; tradeoffs between systems can be assessed and architecture measures can be evaluated which can support cost-benefit analyses and quantitative acquisition decisions (Garcia, 2007).

Throughout the acquisitions life cycle and throughout the lifecycle of the product, executable architecting maintains its importance through configuration management. Past research has identified objectives of executable architecting as: determine the contribution of a system to overall effort, identify blocked resources and provide for alternatives for system development, identify bottlenecks within the process and or network, estimate optimal process times and identify operators, systems or nodes in the overall system that are overloaded and re-distribute activities where appropriate (Garcia, 2007). In essence, executable architectures have the potential to provide a dynamic analysis and insights into behavioral aspect, systems interactions, performance measures, integration difficulties and even exploitable system communications areas.

Deriving Executable Architectures from DoDAF

There have been several modeling techniques for executable architectures identified in past research. A lot of it is mathematical; however, a few software tools have been built to provide analysis of executable architectures as well.

Modeling theory and techniques.

The first analysis technique discussed involves using a form of spectral graph theory. From spectral graph theory, the Perron-Frobenius Eigenvector (PFE), which provides a measure of network effects through the success of each element to the communication cycle (Griendling & Marvis, 2011). The PFE value is summarized to assist in identifying vulnerabilities in networks by identifying the highest centrality. Furthermore, the Coefficient of Network Effects (CNE), which is the ratio between the PFE and the number of nodes in the network, has been identified as a useful measure for efficiency in a network as well as identifying bottlenecking within it (Griendling & Marvis, 2011). For this type of analysis, the SV-1 and SV-2 were identified as the appropriate views, because they convey communications between nodes.

A Markov Chain is a discrete random process with a state space that undergoes transitions from one state to another, depending only on the current state, and not on any other state prior. In other words, the next state only depends on the current state, and doesn't take into account any past states or past transitions. Utilizing Markov Chains, one is able to calculate the probability of future states, given a known initial state. OV-6 and SV-10 products were identified as appropriate views to support Markov Chains (Griendling & Marvis, 2011). From views and products, the state space behavior can be dynamically studied and require little information (Griendling & Marvis, 2011).

Other modeling techniques discussed in past and ongoing research for executable architectures are Discrete Event Simulations (DES) and System Dynamics. DES use numerical analysis to analyze the system (Griendling & Marvis, 2011). Bornejko et al. (2008) utilizes DES to evaluate the OV-1, OV-2 and OV-5 diagrams and supporting views, for the purpose of demonstrating how architectural analysis can evaluate military worth in a system. The OV-5, OV-6 and SV-10 could be used for DES modeling techniques. System dynamics is a technique for modeling and simulating behavior of complex systems and processes (Griendling & Marvis, 2011). Here an SV-4 is appropriate for system dynamics modeling, because it provides a flow of data and between the systems functions, users and sources (Griendling & Marvis, 2011). Monte Carlo simulations were utilized by Eller et al. (2008) to determine the probability of mission success. Here Eller et al. (2008) describes a Monte Carlo simulation using the

OV-5 activity model, now the OV-5b activity diagram, and the OV-2 Operational Node Connectivity Description, now the OV-2 Operational Resource Flow Description. Similar research was also accomplished by Dietrichs et al. (2006)using the OV-1, OV-2, OV-5, and OV-6a viewpoints.

Colored Petri Nets.

Introduced by Dr. Carl Adam Petri in 1962, Petri Nets are a graphical and mathematical modeling tool. Introduced for concurrent processes, Petri Nets have since expanded to higher level forms, one in which we have evaluated is the Colored Petri Net (CPN). Petri nets can be used to model discrete-event systems, distributions for statistical analysis on a system and timing analysis for performance of that system (Beal, Hendrix, McMurray, & Stewart, 2005). The basis of CPNs is to model concurrent systems in a combination of petri nets and modeling language. Typical applications of CPN models, as listed by Kurt Jensen and Lars Kristensen, are communication protocols, data networks, distributed algorithms, embedded systems, business processes and workflows, manufacturing systems, and agent systems (K. Jensen, 2009). CPNs have the ability to model time between events, as well as for individual packets of information through forms of automatic simulation. CPNs also allow for a more interactive modeling in which the modeler is in control of each step, allowing for various scenarios to be observed in detail and the effects of a single step to be analyzed (K. Jensen, 2009). State space analysis and performance analysis are also among the capabilities of modeling and simulation in a CPN (K. Jensen, 2009). An example CPN model for a simple protocol, created by Marc Jensen of Aarhus University in Denmark for CPN tools is shown below:

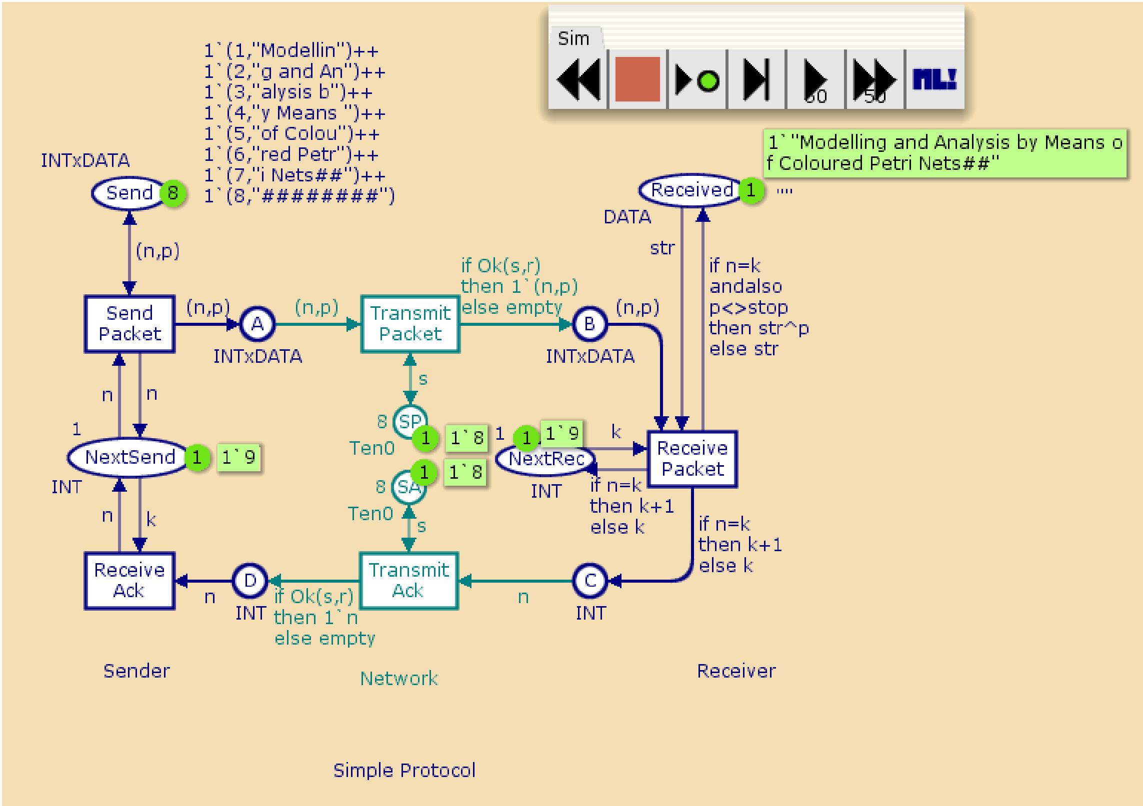


Figure 2. Example CPN of a Simple Protocol

The basics of a CPN model are places (ellipses or circles), transitions (squares), arcs and tokens. CPN modeling and simulation has been documented by many sources as a way to create and analyze executable architectures. Viewpoints OV-6 and OV-5 have been identified as DoDAF products to produce the CPN executable architecture, however, still more information is needed. This information includes scenarios, initial conditions, additional rules and system properties not identified by DoDAF (Griendling & Marvis, 2011). CPNs are also not without faults, they fail to easily allow for an adaptive environment to be modeled. Timing between states can also not be specified which doesn't allow for temporal effects to be considered (Mittal, 2006).

Simulink.

Simulink® is an environment for multidomain simulation and Model-Based Design for dynamic and embedded systems (MathWorks, 2012). The software can also host a wide variety of plug-ins, ranging from RF simulation tools to state machine and flow charts. The tool is typically used to run continuous, discrete, or triggered event simulations. The elements used in Simulink have a close relation to SySML/UML entities, making the mapping of DoDAF elements to Simulink workspace feasible. In the article by Carl-Johan Sjöstedt (Sjostedt), a simple relationship table between Simulink concepts and Unified Modeling Language (UML) elements were created, shown in Figure 3.

Simulink concept	**UML 2** concept
Primitive block	Class
Subsystem block	Class, containing properties corresponding to contained blocks
Line	Connector
Branch	Connector
Port	Port

Figure 3. Structural Concept Mapping

Because of the wide range of elements that Simulink can model and simulate, it can be used for complex systems of systems, where many different subsystems may interact. While Simulink can analyze many different aspects of a system, its ideal function would be to simulate system lags across various nodes. This function can find system bottlenecks, delays and opportunities for maximizing efficiency. A disadvantage of using Simulink for DoDAF executable architecture is that there is a lack of previous research in the field available.

Selecting the Tool and Potential Analysis Techniques

Many software platforms were identified in research as potential tools to create executable architecture from DoDAF including: ViTech Core, IBM Telelogic System Architect, Rockwell Automation Arena, Proforma ProVision, CPNtools, MATLAB/Simulink and Excel Add-ons. Given time and resource constraints, only MATLAB/Simulink and CPNtools were assessed. After weighing the different options for software platforms, Simulink was ultimately chosen as the tool for this study. As stated before, its similarities to SysML/UML allow for easy translations from DoDAF to the Simulink workspace. The flexibility ensured the proof-of-concept could be presented for a variety of case studies. Finally, because Simulink has been used widely in industry and universities for many years, there is an abundance of tutorials and example models available to the public allowing for easy familiarity for the software and toolboxes. Table 1 below describes the decision matrix the led us to select Simulink over CPNtools.

Table 1. Software Platform Selection Criteria

Criteria	CPNtools	MATLAB/Simulink
Previous research found as a tool for EMBSE using DoDAF	Several previous research studies	None
Use in industry	Some	Extensive
Personal familiarity	None	Moderate familiarity with MATLAB
Ease of use	Training required	Training required
Flexibility	Little	Extensive
Analysis	Limited	Unlimited
Executable (from DoDAF)	Yes	Yes

Potential techniques for analysis in Simulink from previous research included a number of different areas discussed in the sections above. The analysis methods that were ultimately selected to use in the modeling assessments in chapter 4 were the Monte Carlo

Method, latency (process delays), Discrete Event Simulation (DES), and risk. Table 2 below shows a summary of all of the methods researched. The methods were selected because they were effective for a proof of concept and could be presented to others with little room for confusion.

Table 2. Analysis Techniques

Potential Areas for Analysis	
Colored Petri Nets	Graphical oriented analysis of communication protocols, distributed systems, work flows, etc.
Markov Chains	State transition analysis for a number of random and independent variable
Latency	A time based analysis to determine various latency through different nodes of a system
Bottlenecking	Analysis to find what nodes in a system's operation has minimum capacity or act as a vulnerability
Discrete Event Simulations (DES)	Analysis through timed sequence of events within a system or process
RF Link Analysis	How can variations in component attributes affect an RF Link (antenna size, power, etc.)
Risk	How will various scenarios effect cost, schedule or performance
Monte Carlo Simulations	Probabilities based on a number of dependent random variables
Perron-Frobenius Eigenvector (PFE)	Provides a measure of network effects through the success of each element to the communication cycle
Combinations of the above	

Remotely Piloted Aircraft Communications Architecture

The development of the *as-is* architecture modeled the current status of the RPA communications across ground, air, and space layers. To build the as-is model (shown in Appendix B), members of the Advanced Concept Division of MILSATCOM gathered information from a number of stakeholders across the DoD including users, mission schedulers, network operators, network authorities, and communications experts. The model was created for several reasons; first to give Air Force leaders a quick look at the state of global RPA communications architecture; and second, to form a taking off point for developing an objective steady-state architecture for RPA communications, known as the *could-be* architecture. The *could-be* architecture was then developed from identified capability gaps in the as-is model. (SMC/MCX, 2011)

Although the OV-1 *as-is* model gives an overview of overall system architecture, other DoDAF viewpoints provide the supporting data required for an executable model. In particular, an OV-5 (Operational Activity Model) is one of the pillar viewpoints to create a simulation. An example of this is provided in Appendix B. In this model a step-by-step of all the steps involved for authorizing and provisioning a network for a given user are shown. These steps are broken out by responsible party and highlight that there are multiple cross-organizations interactions involved. Although it is a DoDAF compliant model, there are still many limitations. From this model it would not be possible to determine how long the full process would take, how long each organization has to respond, if there are any data mismatches, and where the best areas for efficiency improvements are. This model in conjunction with other DoDAF viewpoints is an ideal candidate to be used for an executable model.

Summary

The conducted Literature Review indicates that the overall goals of the DoDAF based executable model is viable, as multiple research papers have already reviewed this topic for previous and current versions of DoDAF (1.0, 1.5, 2.0). This review allows us to consider the tools, modeling techniques and theories which are applicable to executable architecting. The main tool of interest from previous studies, CPNs, was found to have a wide range of research and application to DoDAF architectures and DoD systems. However, due to the limitations imposed by the software for analysis, and the lack of familiarity among engineers, Simulink was chosen to be the only software tool evaluated. Simulink, a customizable tool, could also be capable of creating a CPN style model. Other tools may exist, and many were found to be in beta stages, thus the reader is referred to the DoDAF web 2.02 for a closer look at the ongoing updates and tools available which directly apply to DoDAF.

The final part of the literature review explored work in the current architecture of RPA communications. The DoDAF models from these efforts are a practical and relevant resource to demonstrate an executable model. The executable models created from these DoDAF products in MATLAB/Simulink will be reviewed for validity and relevance. In the following chapter, the methods and techniques derived from the literary review will be formulated into a plan and approach to build and analyze executable architectures in Simulink.

III. Methodology

Chapter Overview

This chapter describes the methodology that was used to conduct the study, develop the results and reach conclusions. A majority of the methodology is studying executable architectures and DoDAF views to figure out how they can be interwoven, if at all. This also included gathering past research as a foundation. The other portion of the methodology lies in deriving and using Executable Model Based Systems Engineering (EMBSE) from actual DoDAF products. This involved finding an executable architecture platform, studying compatibilities and building the executable architectures within this platform. It also involved gathering DoDAF views and breaking them down into their executable parts, as well as creating and using supporting DoDAF views that were not provided. This section will also describe how the results of this study were presented to a selection of system experts from both Systems Engineering and RPA Communications fields to validate both the method and results based on a set of standard evaluation criteria.

Approach

The following list describes the actual approach that was taken for the study, results and finally the analysis for this thesis. It is important to note that a large portion resides in understanding DoDAF, executable architectures as well as Simulink as an environment for DoDAF executable architectures. A significant amount of time was spent investigating and attempting to use executable software tools, such as the aforementioned

Colored Petri Nets (CPN) tool, for viability. The outcome of the studies, further described in the results section, allowed for executable architectures to be built from a foundation of DoDAF Views. These outcomes were presented to the system experts for conclusions to be drawn on the thesis objectives.

As the first step in our study, a significant amount of time was spent becoming familiar with the concepts used in this research effort. This included, studying and understanding DoDAF, executable architectures and the executable architectural tools. Additionally we needed to become proficient at MATLAB/Simulink, the platform used to prove the concept.

The next step in our study was to build the initial models using the research described in chapter 2 of this thesis. This involved the developing UML like executable models, and mapping UML properties to Simulink functions. We then developed the models and analysis in Simulink, using real DoDAF views from the MILSATCOM systems. Upon completion of the models, we ran the simulations and analyzed the results. The research, the models and the results were then presented to knowledgeable MILSATCOM system acquisitions members. From there comments and questionnaire results, conclusions on the thesis objectives were developed.

Executable Architecture for Analysis

The premise of this study is to show how DoDAF can be used as a way to provide EMBSE to assist analysis efforts. This study attempts to show how current DoDAF architectural products can be made executable and analyzed. The results attempt to demonstrate the viability of utilizing available software such as MATLAB/Simulink, and how to convert between the common languages SySML/UML used in DoDAF and the Simulink modeling language. The following figure displays the suggested path we developed for analysis of DoDAF products:

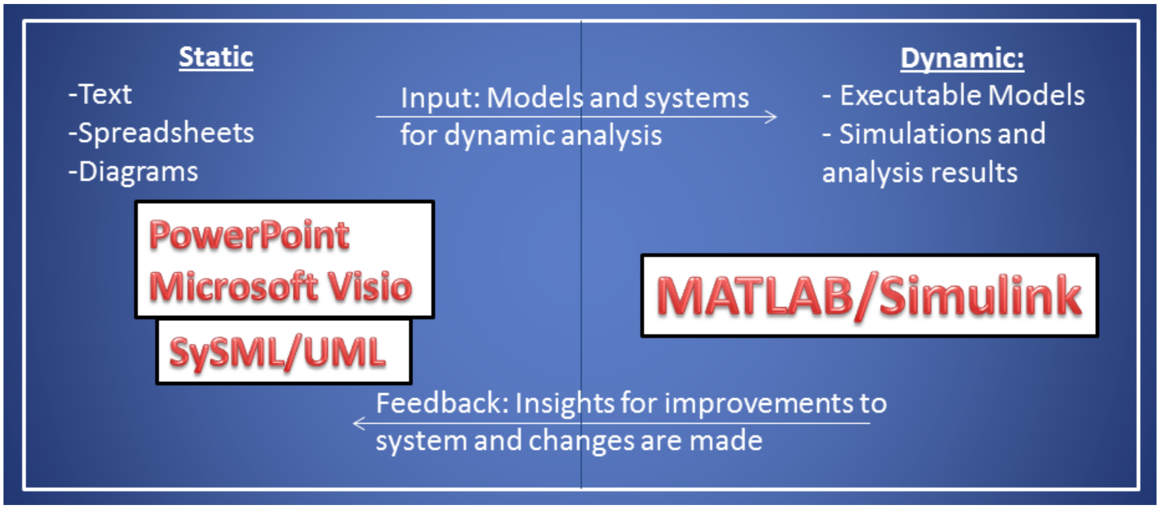


Figure 4. Analysis of DoDAF Products

DoDAF Products.

The following DoDAF products were used to create and support the modeling accomplished in Simulink. With the exception of the Overview and Summery Information, each of the following DoDAF products can themselves be represented by a Simulink model or represented within the model. For example, the OV-6a rules model can be represented within the OV-5b activity diagram through the constraints or rules in which the executable model behaves. Each diagram described represents a significant aspect of the system and system of systems for a given Department of Defense product and was either used to build the executable architecture, or was used to provide supporting information. These architectures were chosen based on their applicability to EMBSE. The viewpoint, a description of the viewpoint and its relevancy to the executable models can be found in the table below.

Table 3. DoDAF Views and Descriptions

DoDAF Viewpoint	Description	Reason for Including in EMBSE
Integrated Dictionary: All View-2 (AV-2)	An architectural data repository with definitions of all terms used throughout the architectural data and presentations.	Using this viewpoint is important in keeping all architecture references and definitions consistent from the original DoDAF to the executable models
High Level Operational Graphic: Operational Viewpoint-1 (OV-1)	This is the high level graphic/textual description of the concept. This can be used as a true backbone to the Simulink model, with all interfaces, resources, actions and data being described by products introduced next.	This study does not model this viewpoint; however, it can be used as a backbone to the executable architecture, or to help ensure you are keeping a model consistent with a larger architecture. A larger executable architecture could begin with this viewpoint and be further defined by rest of the viewpoints.
Operational Resource Flow Description: OV-2	This is a diagram which describes the resource flows exchanged between operational activities. This is a diagram that will be modeled in Simulink.	Similar to the OV-1, this isn't modeled directly and can be used for the backbone of an executable model for analysis. An executable model could describe the resource flow efficiency.
Operational Resource Flow Matrix: OV-3	The Operational Resource Flow Matrix details Resource Flow exchanges by identifying which Operational Activity and locations exchange what resources, with whom, why the resource is necessary and the key attributes of the associated resources.	The OV-3 has been used for the process delay Model Assessment discussed in Chapter 4 and is crucial because it contains the temporal relations of each of the transitions and activities in the executable model.
Operational Activity Model: OV-5b	This is a diagram that describes the context of capabilities and operational activities and their relationships among the activities, inputs, outputs, performers and data objects. This diagram will also be used as a model in Simulink. This diagram is an activity diagram in UML and is further broken down by OV-6a/c models.	The OV-5b was chosen for process delay and discrete even analysis based on directions from previous research. It also almost directly translates to an executable model in Simulink and forms the backbone of the process delay model described in Chapter 4. This architectural model has potential for many variations of analysis because of its easily executable nature and relation to the overall concept of operations for the system.
Operational Rules Diagram: OV-6a	This is one of three models used to describe the operational activity. It identifies business rules that constrain operations.	The OV-6a supplements the other viewpoints by adding constraints and rules for any node that can have more than one outcome or direction.
Event Trace Description: OV-6c	This is a diagram which is the same as the sequence diagram in UML. This is another model used to describe the operational activity. It traces actions, or sequence of events, in a scenario or activity.	This model can be used to further break down the OV-5b diagram in Simulink. A single activity can be broken down into a subsystem of events.

System Resource Flow Description: SV-2	This is also a diagram which identifies the resource flow exchanged between the systems. This diagram differs from the OV-2 in that it is systems specific and leaves out the other actors or personnel involved. Depending on the type of modeling and level of detail desired, a SV-2 may be sufficient for a simple systems modeling in Simulink.	The SV-2 is useful in defining nodes in a Simulink model and which other nodes or subsystems they will interact with. Other viewpoints are required to create an executable model
SV-6: System Resource Flow Matrix	Provides details of system resource flow elements being exchanged between systems and the attributes of that exchange.	The SV-6 defines the information exchanged between interfaces of the nodes in the SV-2. The information combined from a SV-2 and SV-6 can define most of an executable architecture
SvcV-9: Services Technology and Skills Forecast	The emerging technologies, software/hardware products, and skills that are expected to be available in a given set of time frames and that will affect future service development.	The SvcV-9 is useful for executable models that incorporate possible future architectures by defining technologies and capabilities for the short, near, and long term

Simulink Modeling from UML.

As previously defined in Chapter 2, Simulink modeling can be used to model behavioral UML diagrams (Use case, state machine and activity diagrams), information and resource flow diagrams, as well as other analysis areas comprised of DoDAF views. Aspects of these are further defined by supporting documentation in interaction diagrams (sequence, communication, timing and interaction overview diagrams). UML is the defining language of the majority of the diagrams used to model in Simulink. Therefore, it is important to convert from UML to Simulink. A use case diagram displays the actors and scenarios, where a single use case can be represented by an activity diagram which in an OV-5b as described above. An activity in the activity diagram is further represented by a sequence diagram, which is an OV-6c as described above. The data flows, states, timing interactions and resources are further defining and supporting diagrams in DoDAF. Activities, attributes, data flows, timing interactions and actors have been linked to portions of the Simulink executable architecture models. These can allow for a dynamic analysis of the DoDAF views in UML language.

Executable model building in Simulink used previous research as discussed in the literary review, as well as adding additional customization as necessary to build complete executable architectures in Simulink. The OV-5b activity diagram, an essential DoDAF viewpoint, was identified as a potential candidate for conversion to executable architecting. This is based on previous research all indicating the analysis benefits of DES, latency analysis, and system dynamics among other potential analysis. A model assessment was formulated to convert it to an executable model in Simulink for analysis. In an effort to further study EMBSE techniques, two additional case studies were created; a Monte Carlo simulation and a cost analysis model. These were based on analysis methods found in the research and DoDAF viewpoints from MILSATCOM systems. Essentially, executable model building began with a simple framework as laid out by AbuSharekh et al. (2007) and Griendling and Marvis (2011), but was expanded upon as necessary for analysis and application to Simulink. Also, the executable models have been created to be applicable to the RPA systems and analysis in which the DoDAF views belong.

Additional tools and resources have been utilized as fit for executable modeling and analysis in Simulink. MATLAB and Simulink have the ability to create a graphical user interface (GUI) as an easy tool to edit system parameters and display results, allowing for an array of customized analysis techniques. Simulink also has various toolboxes for modeling and analysis that have been explored as applicable to types of executable architectures created. Simulink models have been created in a variety of ways to show effectiveness in creating and analyzing architectures, as well as the breadth of customization and adaptability.

Study Experts

Study experts from both Systems Engineering and RPA Communications fields were briefed and shown a demonstration of the finished executable architectural products. The brief covered the objectives, methodology, a brief description of DoDAF and Simulink and the results of the creation of the executable architectures. These experts were allowed to use, run and change parameters of the Simulink EMBSE examples. Afterward they were given the opportunity to fill out a standardized survey containing the questions addressing aspects of the thesis objections as well as their own familiarity on the topics. This survey can be found in Appendix A.

The involvement of the systems experts allows for development of a *value added* conclusion, as well as a confirmation of the executable models that have been built. Experts will give insights into the potential benefits for current and future DoD systems, allowing for continuous research or use of executable models. Expert feedback will also validate the accuracy of the models and the benefits of EMBSE using DoDAF which we are investigating through case studies.

A total of 10 experts participated in the study. They covered a wide range applicable areas of interest to our research, including software developers, systems engineers, and project managers. All of these experts work in a MILSATCOM related field, an important criteria for meaningful feedback. Survey results and general feedback from these briefings are found in chapter 4

Summary

Executable Models were created in Simulink from DoDAF products provided by the advanced concepts division of MILSATCOM and then evaluated by experts. DoDAF models that cannot be provided by this division of MILSATCOM will be created for the purpose of this study. Methodologies discussed in this section will be used to create the executable models from a selection of test case DoDAF architecture products. The results from the creation of the executable models, results from the executable models themselves as well as results from the study experts are presented in the following chapter.

IV. Analysis and Results

Chapter Overview

This chapter covers the final products and results of the previous methodology. Previous tools utilized in past studies were found to be useful for specific types of analysis, while Simulink allowed for executable architecting as well as analysis and flexibility. MATLAB/Simulink combines and compliments many of the identified areas of analysis for executable architectures as well as being a common and well known tool already used across many disciplines of engineering. Simulink was the sole tool used in the study and creation of executable architectures and results presented to the study experts. DoDAF architectural views were able to be converted from UML to Simulink and made to be executable. The views were also able to be used to create Simulink executable models which could be used to analyze the systems in question. Results from the executable models as well as the expert evaluations will be presented. Analysis of the results will be used make conclusions and specific recommendations discussed further in chapter 5.

Results of Executable Modeling:

Model Assessment 1: Operational Delays.

The first executable models created in Simulink were based on Figure 19 OV-5b Provide Satellite Access Authorization in Appendix B, created by Sam Griffin from the Engineering Division of MILSATCOM. The OV-5b activity diagram has been found to provide the basis for a discrete event simulation (DES) analysis of the system or process being modeled. DES was used to provide analysis on the operational delays in the process being modeled.

Other DoDAF viewpoints were not originally created as part of the Acquisitions process or were not provided to us due to classification concerns. We had created them ourselves as required for the purpose of this thesis to fully define the executable architecture. The OV-6a operational rules model was created to illustrate the constraints and how to handle decisions that lie within the executable model. The OV-3 resource flow diagram was added to define the temporal aspects of the executable model, but also defines what the data is that is flowing through the executable model at each point. The viewpoints can be found in Appendix B. The AV-2 is the integrated dictionary where all the definitions of the terms used throughout the products can be found. The below diagram shows the DoDAF models that were found to be useful for a DES analysis on the process delays.

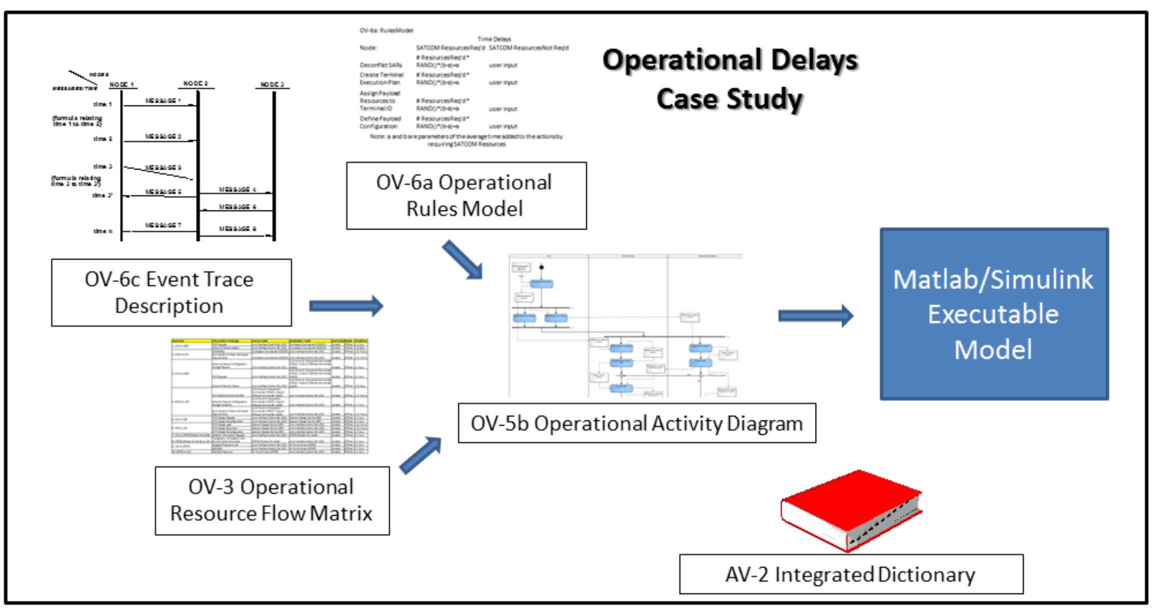


Figure 5. DoDAF Viewpoints used for Model Assessment 1

In this model assessment, there were two versions of the Simulink executable model created. The first Simulink model and associated GUI are shown in Figure 6. System Latency Model. To run this model you first input the various process delays for different activities in the system into the GUI, shown in the input column. After running the simulation the model will return the aggregate process delays at various points throughout the model, shown in the results column. This executable model shows that MATLAB coding and standard Simulink blocks alone can be used to convert a DoDAF view into executable analysis and results. However, this model uses continuous non-discrete time based signals that don't focus on the activities. Transport delay blocks were used to represent the activities in this model.

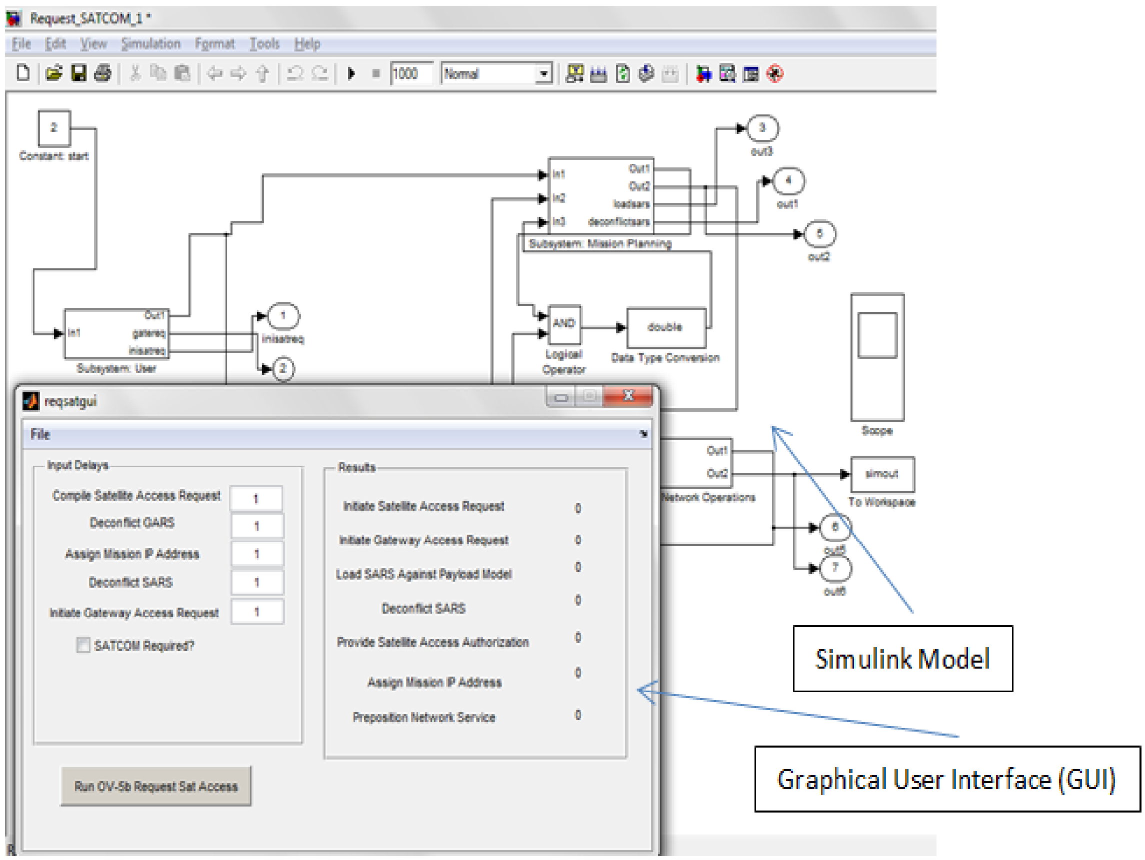


Figure 6. System Latency Model

After additional research on modeling DES in Simulink, a toolbox SimEvents was found to provide a solution for creating models for DES analysis. A second model was then created with a trial version of the SimEvents toolbox. This can be seen in Figure 7 below. More figures can be found in Appendix E. Server blocks allow for modeling the activities themselves in an event based executable model, providing statistics outputs, where the servers act as events which take an X amount of time. With this toolbox, an executable model was able to be created that more closely resembled the DoDAF OV-5b view. The DES analysis allowed for a multitude of results. These results included, the amount of authorizations processed in a given time period, the amount of time a single authorization takes to proceed through the process, how many authorizations are being processed, how many are backed up and the average wait time for an authorization to begin processing. Using a queuing block, we are also able to visualize the authorizations being processed, or held up. This DES analysis could have a multitude of other potential results pertaining to the operational delays, such as bottlenecking. Ultimately, the SimEvents version of the OV-5b executable model was presented to the study experts as it allowed for the most applicable analysis of the architecture.

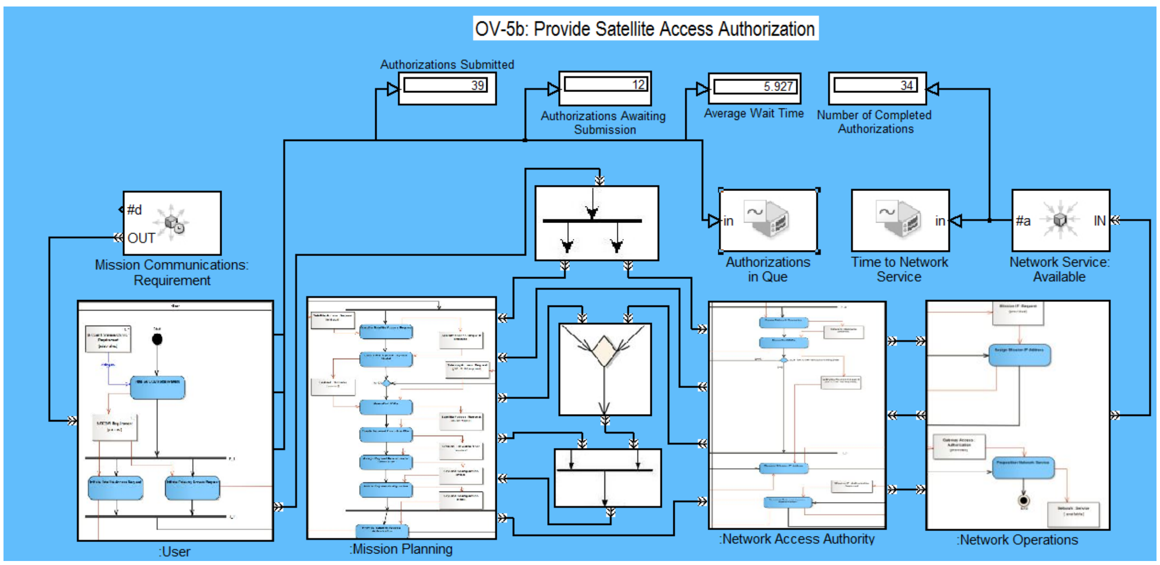


Figure 7. OV-5 DES Model in Simulink

The activity diagram chosen had only a single decision branch and therefore only yielded two possible paths. Path 1 would be where SATCOM resources are required and Path 2 would be SATCOM resources not required. Utilizing hypothetical parameters shown in the OV-3 in appendix B, the program was run to show the different results from the DES analysis for a 72 hour period, with mission communications requirements for satellite access occurring uniformly between .1 hours and three hours. In this 72 hour period, 51 mission communications requirements needed satellite access authorizations. Path 1 allowed for 39 of them to be submitted, taking 4.8 hours to network service available, 12 still were waiting to be submitted with an average wait time of six hours and 34 had actually achieved network service. In the figures below, Figure 9 shows that after 20 hours, the process begins to lag and authorizations begin to stack up. Path 2, where no SATCOM resources were required, allowed for all 51 to be submitted, with only 3 at most stacking up in the queue, 48 total accomplishing network services, and the time to network service being was 4.1 hours. The graph in Figure 10 below shows the Authorization submissions for the second path.

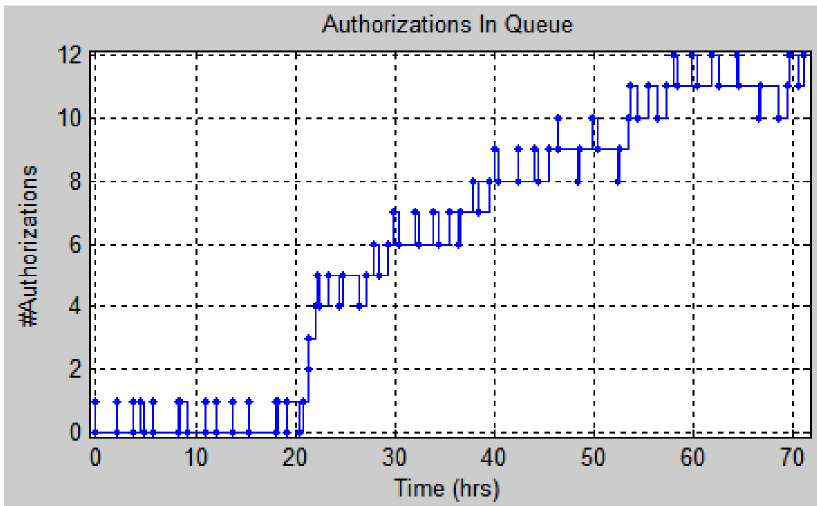


Figure 8. Authorizations in Queue

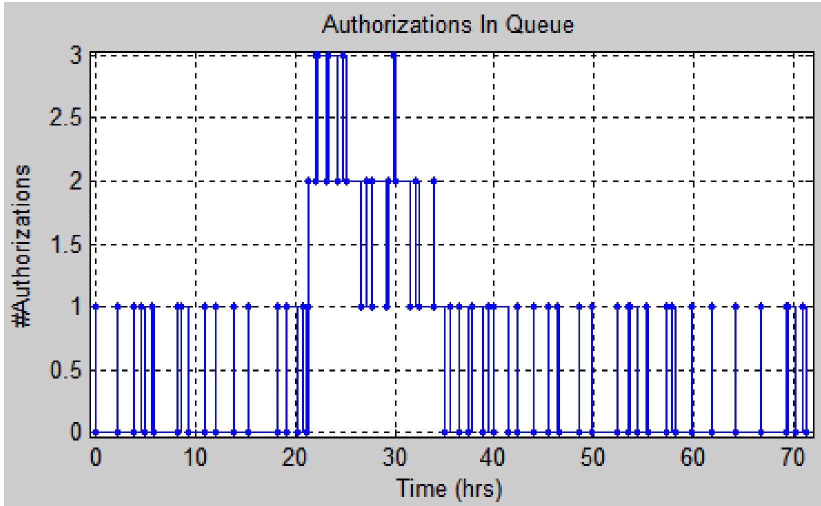


Figure 9. Authorizations Submitted

The OV-5b was able to be converted successfully into an executable model Simulink; however, it was found that the OV-5b provided the backbone, but did not provide all the constraints, rules and temporal definitions as needed by the executable architecture to be fully defined. Other DoDAF viewpoints were required to fill in gaps and add further value to the Simulink model.

Also, the executable models were able to identify a flaw in the OV-5b *Provide SATCOM Resources.* This may have been a mistake in the drawing of the architecture, or the understanding of the UML nodes. When executing the OV-5b in Simulink, the simulation did not continue past the join node when the decision was such that SATCOM resources were required at the decision node. This was due to a *yes* decision which led to a merge node on the same path in the Mission Planning swim lane, thereby leaving the join node with only one input. In a join, by definition, all inputs are required before the activities can continue past it and the executable model was created to emulate the properties of the activity diagram as described by UML, including the join. There could be many interpretations of this flaw, i.e. if the answer is *yes* does that mean there is extra work for mission planning, or if the answer is *no* does that mean there is no need for that part of the mission planning process? For the purpose of this thesis, a work around was created in the executable models, where a *yes* led to a new path in the Mission Planning swim lane, with a merge of the *yes* and *no* paths prior to entering the join. In a merge, activities may continue, even if only one input has arrived. This way the executable model could still emulate the activity diagram without changing the properties of the nodes.

Model Assessment 2. Communication Interruptions.

The second model assessment model produces the number of times an RF link would be lost based on a small probability of weather or intentional jamming interference. The approach to this model is shown below in Figure 10. Model Assessment 2. The SV-2 Systems Resource flow (appendix B) describes each of the nodes in this architecture and what each node interfaces with. Each of those interfaces is defined by the SV-6 System Resource Flow Matrix (appendix B) and is in this case required to make the model executable. The OV-1, Operational Concept Graphic, provides supplemental information to the executable model. The AV-2 is used again in this case to ensure consistency with nomenclature used in both the DoDAF and Simulink models. It is important to note in this example that two outside inputs were included in the model, labeled *outside vulnerabilities*. These two inputs were the probabilities for weather or jamming interference. This is further discussed in Chapter 5.

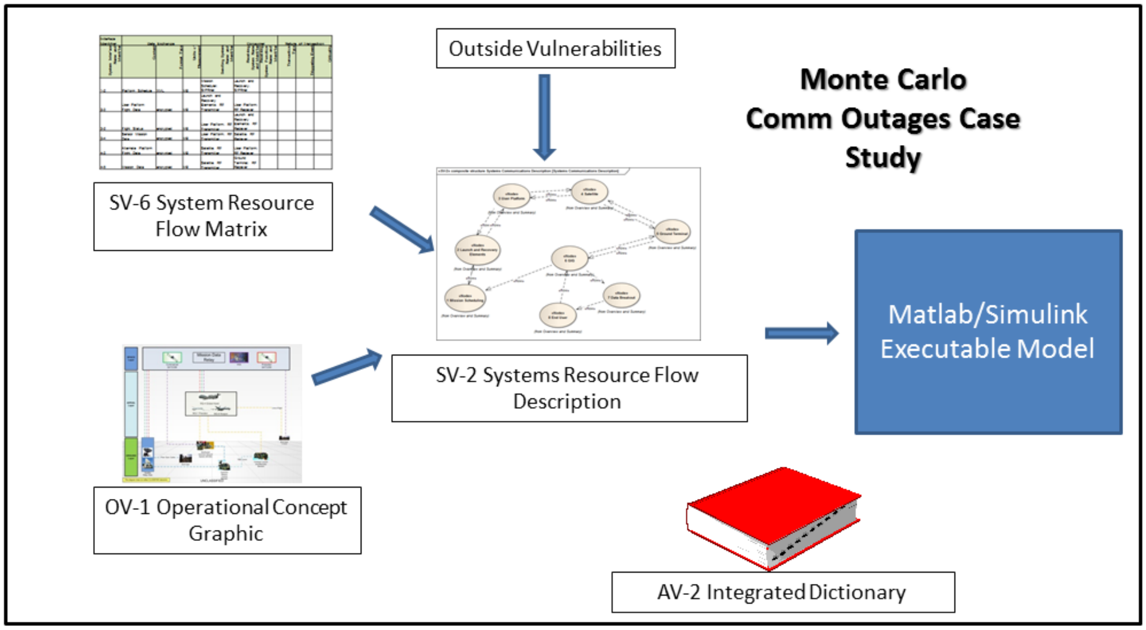


Figure 10. Model Assessment 2

The GUI for this model, Figure 11, allows you to change the number of simulations to run, as a Monte Carlo simulation requires multiple iterations. Probabilities for jamming, weather, average number of sorties per simulation, and architecture changes can be edited in the Simulink file.

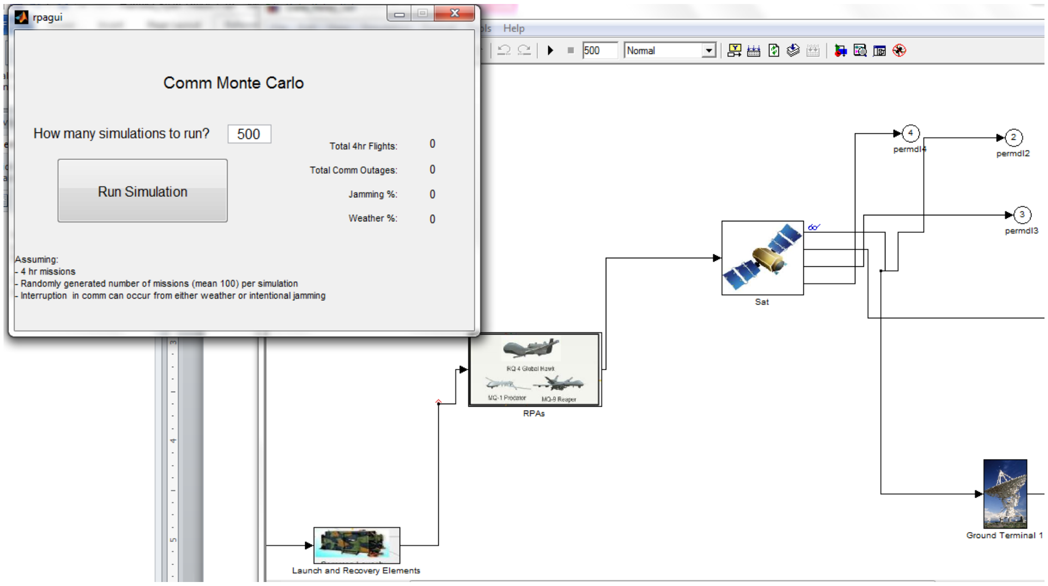


Figure 11. Monte Carlo GUI

Output from this model, Figure 12, is a plot of the number of outages per the number of sorties in that simulation. This data can be exported to excel or analyzed using built in functions in Simulink such as linear or quadratic fitting.

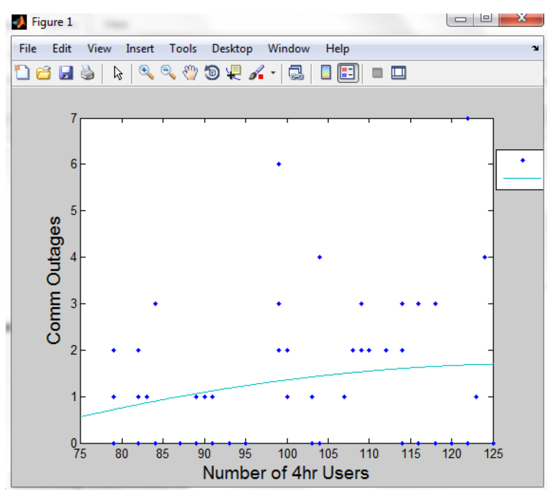


Figure 12. Model 2 Output

The Mote Carlo Modeling Assessment demonstrated that it is effective to add randomness into executable architectures. This concept would be best applied to systems that do not have fully defined parameters or expected outcomes that have not been identified and validated. This capability in Simulink allows insight into system variability and outcomes not otherwise captured.

Model Assessment 3: Cost Analysis.

The third model assessment was design to analyze yearly costs of leasing commercial SATCOM versus costs associated with launching a new military owned satellite. This could be useful in deciding future architectures of MILSATCOM.

Figure 13 below shows the approach and DoDAF used to create this model.

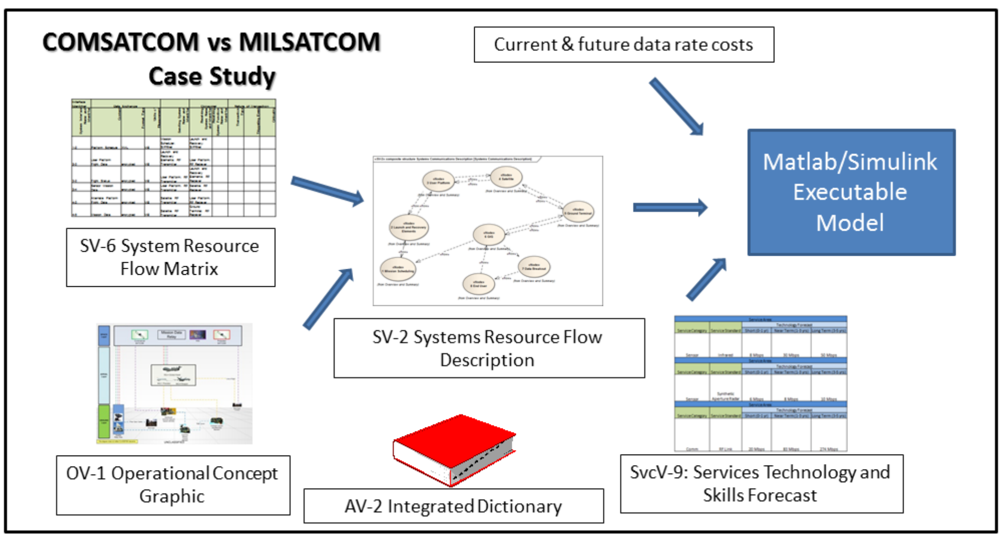


Figure 13. Model Assessment 3 Approach

This model is based on the same background architecture as the Monte Carlo Model, with an addition of the SvcV-9: Services Technology and Skills Forecast viewpoint. The SvcV-9 viewpoint defines technology estimates for the short term (0-

1yrs), near term (1-3yrs), and long term (3-5yrs). In this case, it allowed for RPA sensor data rates to be estimated for use in the simulation.

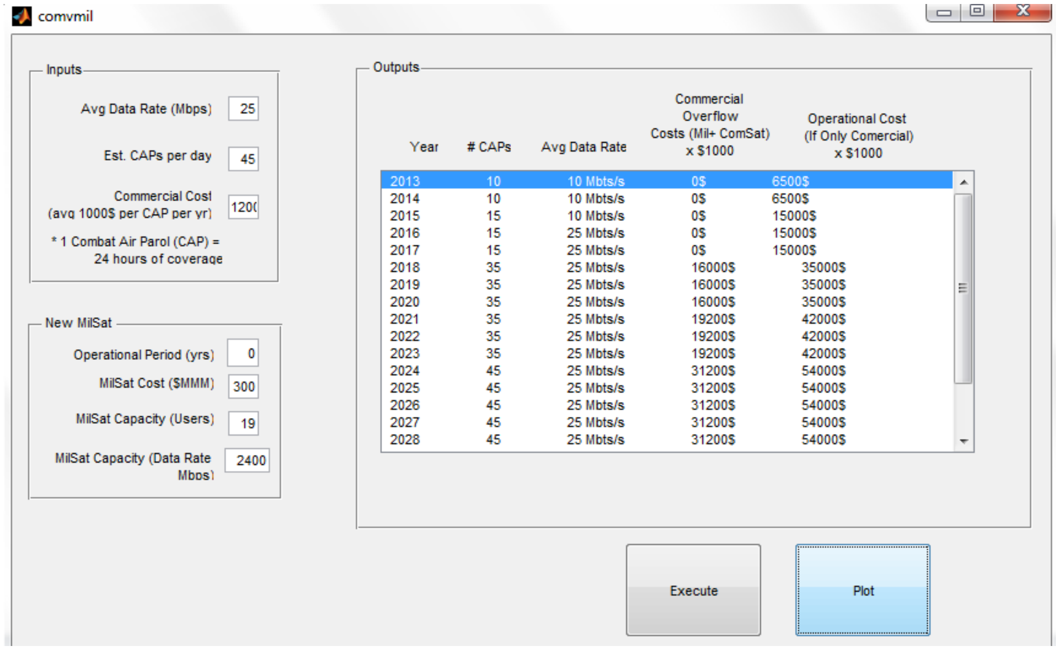


Figure 14. Cost Model GUI

To run the GUI for this model, inputs for the lifetime of the analysis are entered year by year. These inputs include average data rate (from the SvcV-9), estimated simultaneous users (CAPs), average cost to lease commercially, operational period, and cost of a new MILSATCOM satellite with data and user capacities. If the data rate or user capacities are exceeded in that year, then the commercial costs of those additional users are shown in the *Commercial Overflow* Column. The *Operational Cost* column shows what the cost would have been for that year if all users were leasing commercial comm. Figure 15 below shows the results of pressing the plot button.

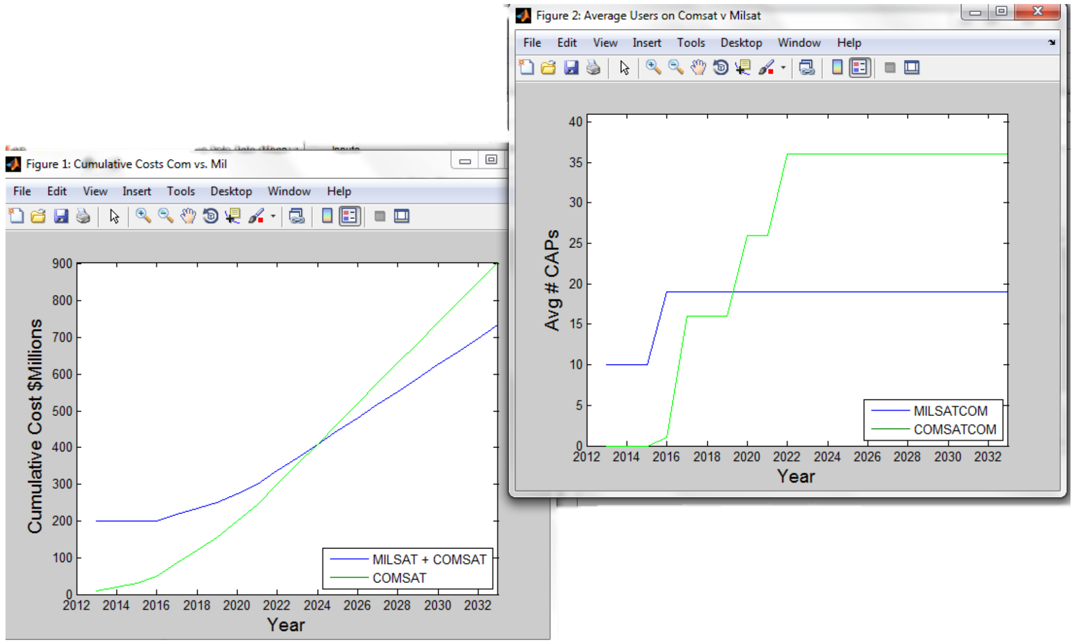


Figure 15. Cost Model Output

The first plot is the total cost by year. The blue line corresponds to the initial acquisition cost of the satellite plus and overflow costs for commercially leased SATCOM. The green line is your yearly cost if all users leased commercial SATACOM. For this example the payoff would have been in about 12 years, in the year 2024. The second plot captures the number of users on commercial SATCOM versus users on the MILSATCOM. The combination of these two would equal the total number of users inputted into the GUI for that year. For this example, the new satellite maxed out its number of users at 19 in the year 2016. After that any additional users are on commercial satellites. An interesting result of this model is that if commercial costs remain relatively constant for leasing SATCOM, than a new MILSATCOM does not pay off. However, if these costs inputted steadily increase around 10% per year you will reach a break-even point in about 10-15 years. Cost increases for commercial SATCOM would be up for discussion on what real world costs will be like in the next few decades. These results should be verified with experts familiar with the systems, discussed further in the next section.

Simulations, like the one presented in Modeling Assessment 3, could be used as a tool for acquisition leaders to determine future system architectures. It successfully represented DoDAF models, such as Svc-9 viewpoint, that allow users to visually see the impact of DoDAF documentation. Potential changes to future architectures can be quickly evaluated and assessed for cost impacts.

Results from the Questionnaire on Study Experts:

Briefing experts in DoDAF, MILSATCOM architectures and MBSE yielded a wide range of feedback ranging from shortfalls to strengths and potential future applications. This feedback was captured via both the questionnaire and verbally during and after the presentations. A summary of the responses is provided below organized by individual model and then overall feedback.

Questionnaire Results.

The survey results showed a very positive trend for executable architectures and Simulink as an environment, while many of the summaries of comments and suggestions discussed a desire for more work to be done in the area. Seven out of eight responses for question 12 *Given your knowledge, the samples and demo provided, would you consider utilizing executable architecting* were answered *Will Consider*, with the other response being *Maybe Consider*. Of those who answered, a majority were also familiar with DoDAF and the RPA systems. Also, 90 percent of the experts answered *Maybe Consider* or *Will Consider* for question seven which asked the reviewer if they would consider

MATLAB/Simulink as a tool to analyze architectures. A majority of the results also showed that the executable models and Simulink environment was between somewhat effective/accurate to largely effective/accurate. Figure 16 summarizes the results for each of the questions pertaining to the thesis objectives (questions 4-12). The question numbers lie along the horizontal axis. The marker for each question is colored according to which type of answer belongs to that question. The marker corresponds to the question's average response, while the bars above and below the marker represent the standard deviation.

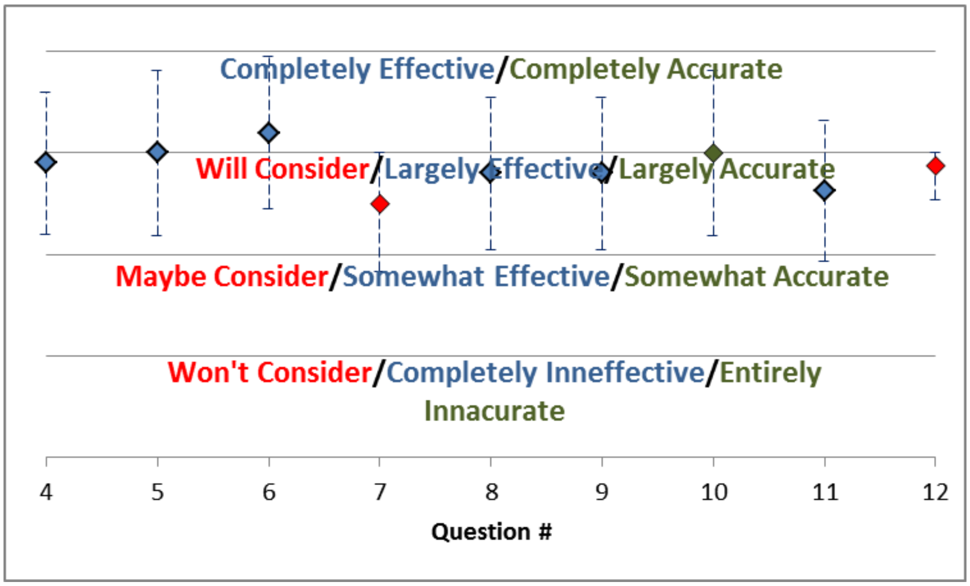


Figure 16. Results by Question

The following tables show the full statistical results for each question of the ten feedback forms administered to the study experts. Questions 5, 10, 11 and 12 all had no answers or need more information marked at least once. Question 10 which asked about the accuracy of the executable model in Simulink to depict the DoDAF model and UML properties may have been worded confusing as 40 percent of the experts choose *need more information* or didn't answer. Of those who did answer question 10, two thirds were familiar with all three, MATLAB, DoDAF and the RPA systems. It is interesting to note that no expert answered completely ineffective in any category of effectiveness for the executable architecture or Simulink as a tool. Table 4 was further broken down by those familiar with MATLAB, DoDAF, RPA systems or all three. This can be referenced in appendix F.

Table 4. Statistical Results from the Questionnaire

	Do you have any prior experience or are you familiar with Simulink/MATLAB?					Q7	**To analyze architectures, would you consider using Simulink as a tool?**				
Code	Value	Frequency	Percent	Total	10	Code	Value	Frequency	Percent	Total	10
1	No Experience	4	40.00%			1	Won't Consider	1	10.00%		
2	Some Experience	5	50.00%			2	Maybe Consider	3	30.00%		
3	Experienced	1	10.00%			3	Will Consider	6	60.00%		
Q2	**With DoDAF?**					Q8	**Are the executable architectures presented effective for evaluating the Systems or System of Systems architecture as described by DoDAF products?**				
Code	Value	Frequency	Percent		10	Code	Value	Frequency	Percent		
1	No Experience	2	20.00%			1	Completely Ineffective	0	0.00%	Total	10
2	Some Experience	3	30.00%			2	Somewhat Effective	4	40.00%		
3	Experienced	5	50.00%			3	Largely Effective	4	40.00%		
						4	Completely Effective	2	20.00%		
Q3	**With the RPA Communications Systems presented in the architectural products?**					Q9	**As presented in Simulink, does this executable architecture effectively represent the DoDAF architectural products?**				
Code	Value	Frequency	Percent	Total	10	Code	Value	Frequency	Percent	Total	10
1	No Experience	2	20.00%			1	Completely Ineffective	0	0.00%		
2	Some Experience	3	30.00%			2	Somewhat Effective	4	40.00%		
3	Experienced	5	50.00%			3	Largely Effective	4	40.00%		
						4	Completely Effective	2	20.00%		
Q4	**Based on the samples and demo provided could the systems architecture be effectively evaluated in an executable environment such as Simulink?**					Q10	**Do the Simulink executable models present an accurate depiction of the DoDAF architectural products just as UML models would?**				
Code	Value	Frequency	Percent	Total	10	Code	Value	Frequency	Percent	Total	6
1	Completely Ineffective	0	0.00%			1	Entirely Innacurate	0	0.00%		
2	Somewhat Effective	3	30.00%			2	Somewhat Accurate	2	33.33%		
3	Largely Effective	5	50.00%			3	Largely Accurate	3	50.00%		
4	Completely Effective	2	20.00%			4	Completely Accurate	2	33.33%		
Q5	**Is the executable architecture effective for allowing a dynamic analysis of the systems architecture it represents?**					Q11	**Is Simulink/MATLAB an effective product for analyzing architectures?**				
Code	Value	Frequency	Percent	Total	9	Code	Value	Frequency	Percent	Total	8
1	Completely Ineffective	0	0.00%			1	Completely Ineffective	0	0.00%		
2	Somewhat Effective	3	33.33%			2	Somewhat Effective	4	50.00%		
3	Largely Effective	4	44.44%			3	Largely Effective	3	37.50%		
4	Completely Effective	3	33.33%			4	Completely Effective	1	12.50%		
Q6	**Has Simulink been effectively used to convert the DoDAF Architectural Products to an executable format?**					Q12	**Given your knowledge, the samples and demo provided, would you consider utilizing executable architecting?**				
Code	Value	Frequency	Percent	Total	10	Code	Value	Frequency	Percent	Total	8
1	Completely Ineffective	0	0.00%			1	Won't Consider	0	0.00%		
2	Somewhat Effective	2	20.00%			2	Maybe Consider	1	12.50%		
3	Largely Effective	4	40.00%			3	Will Consider	7	87.50%		
4	Completely Effective	4	40.00%								

Table 5. All Results for EMBSE and Analysis in Simulink Questionnaire

All Results												
ID#	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Q11	Q12
DH1	2	1	1	3	2	4	3	4	4	4		3
MR2	2	3	2	4	4	4	3	2	3	3	2	3
NN3	2	2	1	2	2	2	2	2	2		2	2
LA4	2	3	2	2	2	2	3	2	2	2	2	3
SG5	1	3	3	4	4	4	3	4	4	4	4	3
LB6	2	3	3	3	3	3	2	3	2	2		
RH7	1	1	3	3	4	4	2	3	3		3	3
DB8	1	2	2	2		3	1	2	3		2	
NY9	3	2	3	3	3	3	3	3	2	3	3	3
NB10	1	3	3	3	3	3	3	3	3		3	3
Average	1.70	2.30	2.30	2.90	3.00	3.20	2.50	2.80	2.80	3.00	2.63	2.88
Stdev	0.64	0.78	0.78	0.70	0.82	0.75	0.50	0.75	0.75	0.82	0.70	0.13

Model Specific Feedback.

Operational Delays Model.

Reviewers of this model expressed interest in how effectively this model mimicked the original OV-5a presented. One reviewer commented that this exact analysis would be helpful on the Control and Planning Segment (CAPS) architecture currently under acquisition. The expert said that CAPS is looking to answer the exact type of architecture trade off analysis that this executable model aims to address. Most of the reviewers expressed they would like to see additional layers of analysis conducted on this model. For example, in addition to queuing feedback, producing information on which nodes are bottlenecks in the process. Some other suggestions for improvements included adding some randomness to each process node, random kickbacks, and inclusion of branching or failure modes.

Communications Outages Model.

Feedback for improvements of the Monte Carlo Model Assessment mostly included adding additional variables as inputs to the model as well as a wider range of outputs, such as consecutive failures. The experts commented that the ease at which you can insert, remove, or edit random variable inputs with Simulink was a useful function; however, they said that it would be a more effective model if it could be used to answer a more specific architecture question or problem.

Cost Analysis Model.

Presenting this model to MILSATCOM engineers sparked some interesting conversations on current tradeoff arguments for MILSATCOM versus COMSATCOM. Reviewers commented that the model would be more useful if it could incorporate additional cost factors such as user terminal upgrade costs. In one case the evaluator entered in some hypothetical numbers they had previously analyzed and the model yielded a much longer pay back than the 10 year payback his previous work had produced. This indicated we needed to identify all of the assumptions that we had used to help improve accuracy.

Overall Feedback.

We received a magnitude of both positive feedback and constructive criticism when presenting to the experts. The best examples of positive responses included: *this thesis offers definitive proof of concept; the relationships among system are well represented and consistent with the models they are based upon* and *definitely value added*. There were also some strong opinions on the overall concept of the research including: *putting architectures into motion based on UML/SySML architectures is* *exactly what is lacking in the space systems engineering environment* and *executable architectures are the future of MBSE*.

In addition to the positive comments, the experts also identified many areas for improvements. One common theme was a need for more in depth analysis. Some responses to this extent included: *more complex and higher fidelity models would be needed to drive actual system designs but this shows a good need for systems analysis and modeling*, *presentation may be more effective if more factors were incorporated into the models*, and *to consider using Simulink I would need to see more maturation*. Comments also indicated the need to attempt this analysis on larger architectures: *yet to be proven for large more complex systems* or *more complex and higher fidelity models would be needed to drive actual system designs but this shows a good use for systems analysis and modeling*.

The study experts were very helpful in suggesting further research to explore post thesis, which will be captured in the recommendations for action and future research sections of chapter 5.

Summary

This chapter covered the final products and results from the methodology presented in Chapter 3. Three case studies were performed to validate the executable architecture concept discussed in previous chapters. Models from these case studies were presented to a variety of experts in MILSATCOM and systems engineering who served as our study experts. Written and verbal feedback from the experts were analyzed and summarized. Comments range from positive to weaknesses of our model as well as gave us ideas for areas to explore in future research. These comments and results will form the basis of our conclusions discussed in chapter 5.

V. Conclusions and Recommendations

Chapter Overview

This chapter covers the conclusions on the research done into DoDAF compliant executable model based systems engineering (EMBSE), conclusions from the study experts, significance of the research and recommendations for implementation and further research.

Conclusions of Research by Objective

1. Demonstrate that an executable architecture can be derived from DoDAF views in Simulink.

The executable models in Simulink were able to have customized variable data inputs as well as outputs. The demonstrations showed flexible models could be created, simulated and analyzed. The ability to imbed MATLAB functions enables EMBSE to support almost any architecture and form of analysis for execution. DoDAF compliant views were utilized to create the executable architectures and analyze models. An interesting note in the creation of the Process Delay model is that an executable model could be created with few DoDAF products, but not fully defined. The OV-6a (rules model), for example, was necessary to define what happens at the decision point *SATCOM Resources Required*. For the communications outages model, additional information was required as well. Some specific analysis areas requiring real world parameters, such as vulnerabilities like jamming or cost estimates for commercially leasing SATCOM, are not accounted for in the DoDAF products. It is not required that all DoDAF views and models be created; therefore, executable models could lack required defined simulation environment unless simulation and execution is the end product goal, or the DoDAF products are complete and the architecture completely defined. Overall, it was found that the Activity Model (OV-5b) is the ideal product to begin building an executable model, while the rest of the architectural products and parameters would support and further define the executable model.

2. **Evaluate the effectiveness of executable architectures in evaluating DoDAF Models.**

The results of the first model assessment, in which a fundamental error in the use the fork, join, decision and merge nodes was discovered, shines light on how the ambiguities of a static architecture can lead to different understandings. By evaluating architectures in an executable environment, the process can be simulated allowing for the architecture to be evaluated objectively. The feedback from the study panel validated the effectiveness of executable models and the desire to utilize them for DoDAF evaluation. The error discovered in the OV-5b model allowed for feedback into the architecture for a revision. This was just a model assessment for a current system, but had this been a part of new system yet to reach milestone A in the acquisitions process, or leave the architecture development stage, this could have allowed for a feedback into the architecture development to eliminate misunderstandings. The experts, who were all members of the acquisitions community, indicated their interest in this benefit.

3. **Determine if Simulink is an effective tool for analyzing DoDAF compliant architectures.**

Simulink models resembled and acted in accordance with the properties of the DoDAF architectures. Analysis was limited to the case studies presented, however, Simulink proved to be a flexible platform for effective and customizable analysis. Study experts commented on utilizing the techniques presented in the case studies for their own projects and adding in more customization for increased analysis capabilities. Creating both DoDAF architectures as well as the executable architectures for EMBSE adds cost, time and uses resources. More research would need to be accomplished to determine the impact on a project if EMBSE in Simulink in parallel with DoDAF architecture creation is utilized. For the purpose of this thesis and based on the study results and research of DoDAF and executable architectures, utilizing Simulink for EMBSE added value to the architectures and the analysis of them for the system.

Significance of Research

Executable architectures as applied to DoDAF have been researched in previous studies, but have often not discussed in detail the ideal environment to build and conduct EMBSE. The results have shown the effectiveness and applicability of executable modeling in a common environment such as Simulink. What's more, the OV-5b can be directly translated into the Simulink environment and executed. This shows the close similarities between Simulink and UML. Other viewpoints, other than the activity model, then add value in such a way to make EMBSE emulate the real world simulation in the Simulink environment. These similarities may make it possible to utilize Simulink as the simultaneous DoDAF building and executing platform.

Furthermore, EMBSE has shown to have real world applications in current DoD systems. One study participant expressed the desire to begin utilizing it in a current program called Control and Planning Segment (CAPS). CAPS is a mission scheduling service under acquisition for the Enhanced Polar System (EPS) program. The first model assessment demonstrated the viability of Discrete Event Simulation (DES) to analyze process latency and capacity optimization. If utilized early on in the acquisition programs of the DoD, EMBSE and the feedback from it could optimize the processes, leading to more efficient and cost effective systems and systems engineering efforts. Lastly, by creating an executable architecture, requirements are fully captured and ambiguities and misunderstandings are eliminated, which could further save time, money and effort in acquisitions of ever more complex systems

Limitations

EMBSE requires a certain level of complete, accurate and well defined DoDAF products. If there is a lack of completeness in DoDAF products, there may be difficulty fully defining executable models. EMBSE in Simulink may not be able to fully model DoDAF as this study only addressed a small subset of Air Force Systems and DoDAF views, and may need further validation in other DoDAF applications. Also, many organizations already model their systems using internally consistent methods and tools. Some of these tools may have already been purchased and in use making organizations reluctant to purchase new tools or expend resources for training and implementation of EMBSE in Simulink.

Recommendations for Action

Based on the results from the study panel and the research into DoDAF and EMBSE, it is recommended that EMBSE be integrated into DoDAF and acquisitions processes early on to allow for requirements capturing and the much needed dynamic analysis. The benefit would be providing objective results and feedback early on in the acquisitions process to allow for a more efficient and cost effective system as well as stakeholder satisfaction when the requirements are captured and simulated. One of the study experts made the comment that EMBSE is worth requesting research dollars from MILSATCOM leadership to pursue further applications and research. This research could then be applied to some of the work that the Engineering Directorate of MILSATCOM is currently doing into modeling Air Force MILSATCOM assets. Lastly it is recommended to the acquisitions community that DoDAF viewpoints, including the OV-5b, be included as CDRLs or deliverables in acquisitions of DoDAF systems. This will ease the process creating EMBSE for future systems.

Recommendations for Future Research

Future research should focus on automation from DoDAF products to executable architecting or simultaneous development to reduce wasted time and resources having to produce DoDAF views in one platform, then in another for executing. More complex and real world Simulink models should be created with systems beginning the acquisitions process to further determine the impact and evaluate the benefits of EMBSE. Incorporating executable architectures into future versions of DoDAF should also be researched and strongly considered.

Summary

Development of executable models in Simulink using DoDAF complaint models is both viable and beneficial. The objectives of this thesis are not far reaching and the results of this research effort can be easily implemented in the acquisitions process.

EMBSE in the Simulink environment has shown to be a possibility in current systems that are being developed. While DoDAF architectural products are often created, they may often be incomplete without fully capturing the requirements. If implemented, EMBSE can capture and evaluate the requirements early on in the acquisitions process.

Appendix A ***Expert Questionnaire***

Name:

Title:

1. Do you have any prior experience or are you familiar with Simulink/MATLAB?

☐ No Experience	☐ Some Experience	☐ Experienced

2. With DoDAF?

☐ No Experience	☐ Some Experience	☐ Experienced

3. With the RPA communication systems presented in the architectural products?

☐ No Experience	☐ Some Experience	☐ Experienced

4. Based on the samples and demo provided could the systems architecture be effectively evaluated in an executable environment such as Simulink?

☐ Completely Ineffective	☐ Somewhat Effective	☐ Largely Effective	☐ Completely Effective	☐ Need More Info

Why/Why Not/Comments: ____

____

____

5. Is the executable architecture effective for allowing a dynamic analysis of the systems architecture it represents?

☐ Completely Ineffective	☐ Somewhat Effective	☐ Largely Effective	☐ Completely Effective	☐ Need More Info

Why/Why Not/Comments: ____

____

____

6. Has Simulink been effectively used to convert the DoDAF Architectural Products to an executable format?

☐ Completely Ineffective	☐ Somewhat Effective	☐ Largely Effective	☐ Completely Effective	☐ Not Applicable

Why/Why Not/Comments: ____

____

____

7. To analyze architectures, would you consider using Simulink as a tool?

☐ Won't Consider	☐ Maybe Consider	☐ Will Consider	☐ Need More Info

Why/Why Not/Comments: ____

____

____

8. Are the executable architectures presented effective for evaluating the Systems or System of Systems architecture as described by DoDAF products?

☐ Completely Ineffective	☐ Somewhat Effective	☐ Largely Effective	☐ Completely Effective	☐ Need More Info

Why/Why Not/Comments: ____

____

____

9. As presented in Simulink, does this executable architecture effectively represent the DoDAF architectural products?

☐ Completely Ineffective	☐ Somewhat Effective	☐ Largely Effective	☐ Completely Effective	☐ Need More Info

Why/Why Not/Comments: ____

____

____

10. Do the Simulink executable models present an accurate depiction of the DoDAF architectural products just as UML models would?

☐ Entirely Inaccurate	☐ Somewhat Accurate	☐ Largely Accurate	☐ Completely Accurate	☐ Need More Info

Why/Why Not/Comments: ____

____

____

11. Is Simulink/MATLAB an effective product for analyzing architectures?

☐ Completely Ineffective	☐ Somewhat Effective	☐ Largely Effective	☐ Completely Effective	☐ Need More Info

Why/Why Not/Comments: ____

____

____

12. Given your knowledge, the samples and demo provided, would you consider utilizing executable architecting?

☐ Won't Consider	☐ Maybe Consider	☐ Will Consider	☐ Need More Info

Why/Why Not/Comments: ____

____

____

13. Additional comments

____

____

____

____

____

____

____

____

____

____

____

____

____

____

____

Appendix B RPA DoDAF Viewpoints

[Figure 17. DoDAF OV-1: *As-Is* RPA Communications Architecture has been removed for distribution purposes. Copies of the image can be obtained from the authors For Official Use Only]

Figure 17. DoDAF OV-1: *As-Is* RPA Communications Architecture

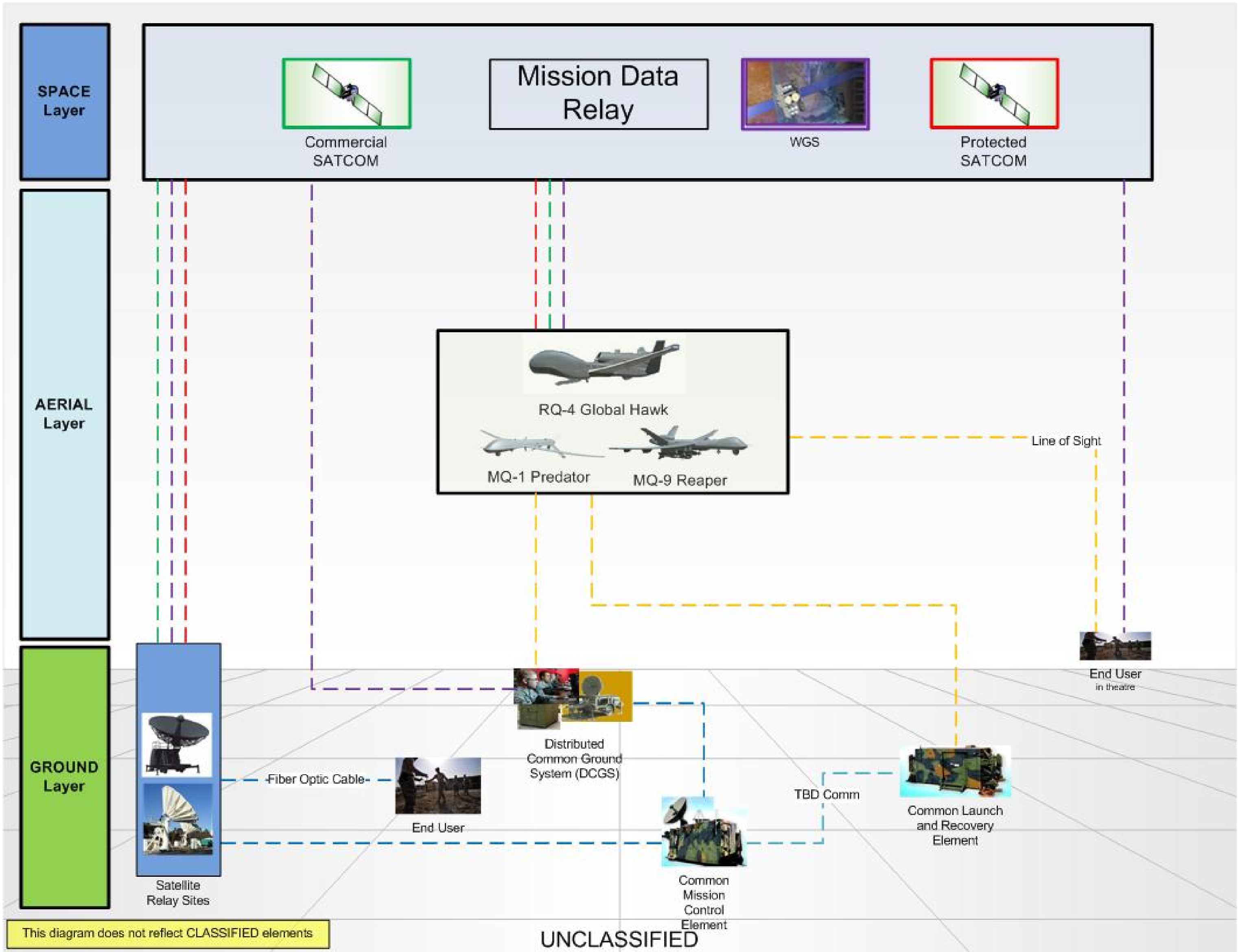


Figure 18. DoDAF OV-1: *Could-Be* RPA Communications Architecture

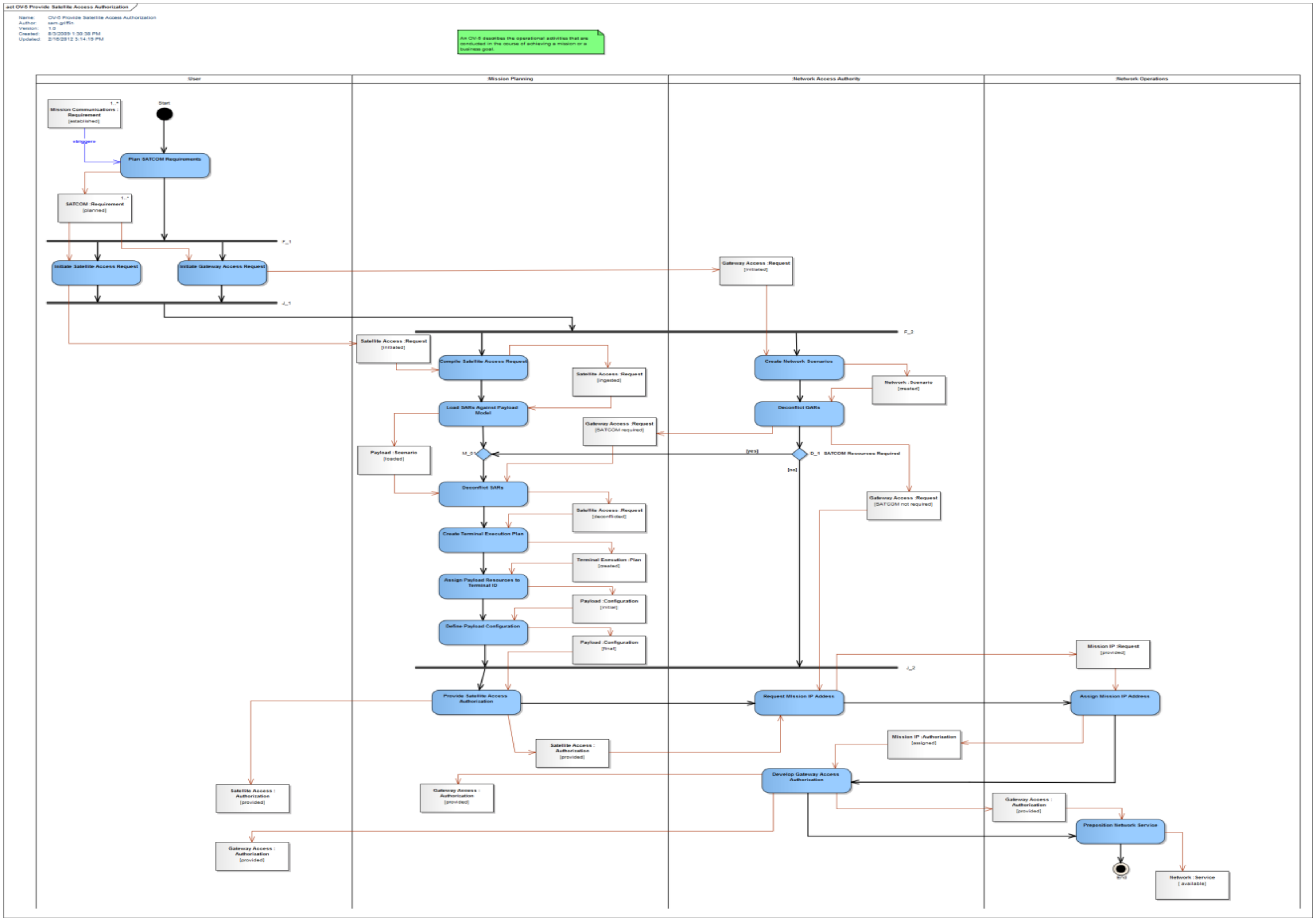


Figure 19. OV-5b Provide Satellite Access Authorization

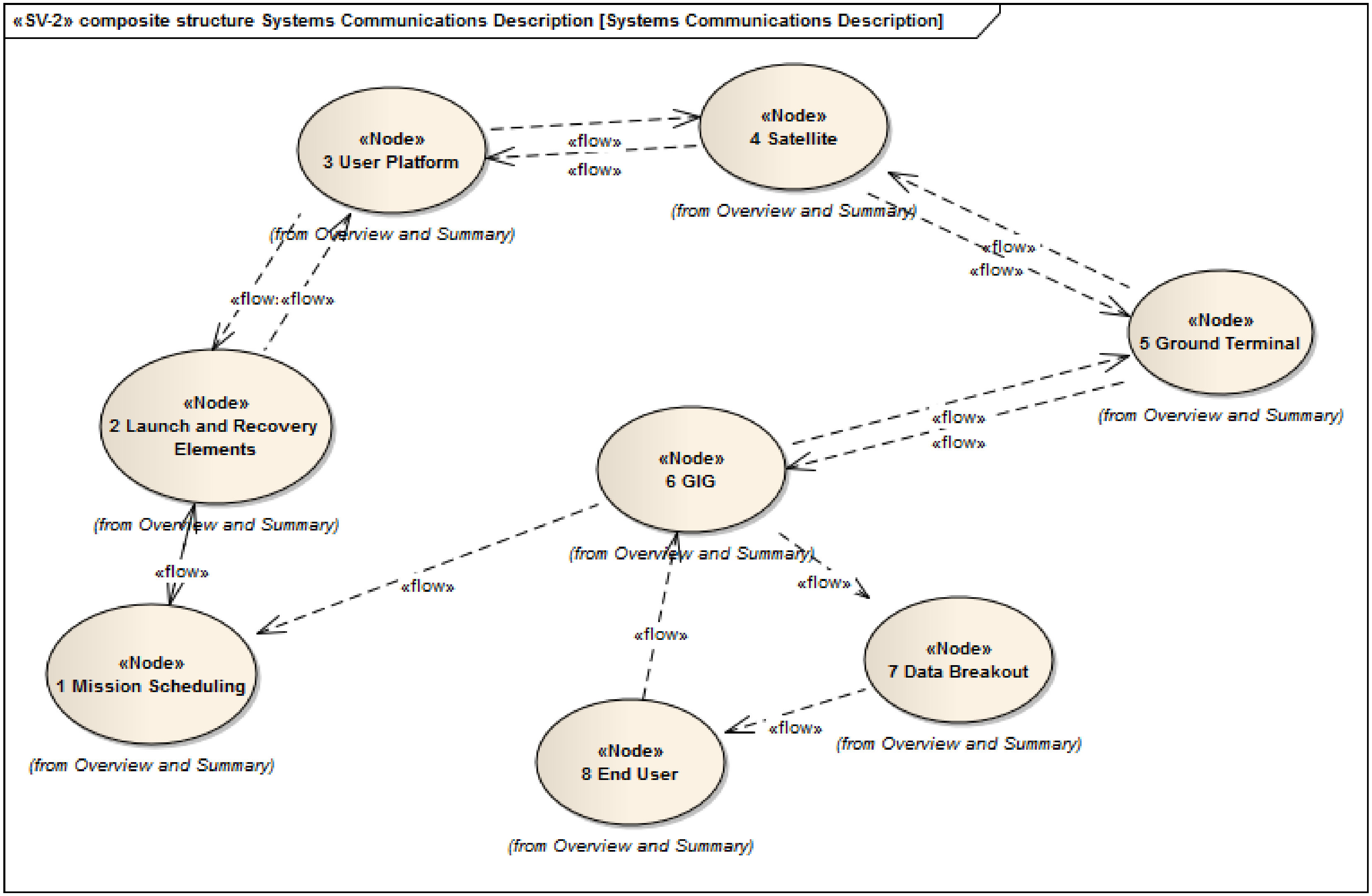


Figure 20. DoDAF SV-2: System Resources Flow Description

Interface Identifier	Data Exchange				Consumer		Nature of Transaction			Performance Attributes				Source	Notes
System Interface Name and Identifier	Content	Format Type	Units of Measurement	Sending System Name and Identifier	Receiving System Name and Identifier	Receiving System Function Name and Identifier	Transaction Type	Triggering Event	Criticality	Periodicity	Timeliness	Throughput	Size		
1-2	Platform Schedule	XML	MB	Mission Scheduler: SIPRnet	Launch and Recovery: SIPRnet										
2-3	User Platform Flight Data	encrypted	MB	Launch and Recovery Elements: RF Transmitter	User Platform: RF Reciever										
3-2	Flight Status	encrypted	MB	User Platform: RF Transmitter	Launch and Recovery Elements: RF Reciever										
3-4	Sensor Mission Data	encrypted	MB	User Platform: RF Transmitter	Satellite: RF Reciever										
4-3	Alternate Platform Flight Data	encrypted	MB	Satellite: RF Transmitter	User Platform: RF Reciever										
4-5	Mission Data	encrypted	MB	Satellite: RF Transmitter	Ground Terminal: RF Reciever										
5-4	Alternate Platform Flight Data	encrypted	MB	Ground Terminal: RF Transmitter	Satellite: RF Reciever										
5-6	Mission Data	encrypted	MB	Ground Terminal: Gov Fiber Network	GIG: Gov Network Connection										
6-1	Schedule Request	XML	MB	GIG: Giv Fiber Network	Mission Scheduling: Gov Network Connection										
6-5	Alternate Platform Flight Data	encrypted	MB	GIG: Giv Fiber Network	Ground Terminal: Gov Fiber Network										
6-7	Mission Data	encrypted	MB	GIG: Giv Fiber Network	Data Breakout: Gov Network Connection										
7-8	Mission Data	encrypted	MB	Data Breakout: VPN	End User: VPN										
8-6	Schedule Request	XML	MB	End User: Gov Network Connection	GIG: Gov Network Connection										

Figure 21. DoDAF SV-6: System Resource Flow Matrix

Need Line	Information Exchange	Source Node	Destination Node	Language	Content	Size/Units	Media	Collaborative	Timeliness	Throughpu	Policy
1, External to USER	Mission Communication Re	External	Plan SATCOM Requirements (USER)	Data/Text	Satellite Access Request Missi	Variable	SIPRnet	Collaborative	Trigger (inst	Variable	MIL-STD
2, USER to Mission Planning	SATCOM Requirement	Plan SATCOM Requirements (USER)	Initiate Satellite Access Request (USER)	Data/Text	Planned SATCOM Requiremen	Variable	SIPRnet	Collaborative	1 Hour	Variable	MIL-STD
	Satellite Access Request	Initiate Satellite Access Request (USER)	Compile Satellite Access Request (Mission	Data/Text	Satellite Access Request	Variable	SIPRnet	Collaborative	0.15 Hour	Variable	MIL-STD
3, USER to Network Access Authority	SATCOM Requirement	Plan SATCOM Requirements (USER)	Initiate Satellite Access Request (USER)	Data/Text	Planned SATCOM Requiremen	Variable	SIPRnet	Collaborative	1 Hour	Variable	MIL-STD
	Gateway Access Request	Initiate Gateway Access Request (USER)	Create Network Scenarios (Network Access	Data/Text	Gateway Data	Variable	SIPRnet	Collaborative	0.15 Hour	Variable	MIL-STD
4, Mission Planning to Network Access Authority/USER	Satellite Access Request	Compile Satellite Access Request (Missior	Load SARs Against Payload Model (Mission	Data/Text	Satellite Access Data	Variable	SIPRnet	Collaborative	0.5 Hour	Variable	MIL-STD
	Payload Scenario	Load SARs Against Payload Model (Missior	Deconflict SARS (Mission Planning)	Data/Text	Payload Scenario	Variable	SIPRnet	Collaborative	0.15 Hour	Variable	MIL-STD
	Deconflicted Satellite Acces	Deconflict SARs (Mission Planning)	Create Terminal Execution Plan (Mission Pl	Data/Text	Satellite Access Data	Variable	SIPRnet	Collaborative	0.5 Hour	Variable	MIL-STD
	Terminal Execution Plan	Create Terminal Execution Plan (Mission P	Assign Payload Resources to Terminal ID (M	Data/Text	Terminal Execution Data	Variable	SIPRnet	Collaborative	1 Hour	Variable	MIL-STD
	Initial Payload Configuratio	Assign Payload Resources to Terminal ID (I	Define Payload Configuration (Mission Plar	Data/Text	Payload Configuration Data	Variable	SIPRnet	Collaborative	0.1 Hour	Variable	MIL-STD
	Final Payload Configuration	Define Payload Configuration (Mission Pla	Provide Satellite Access Authorization (Mis	Data/Text	Payload Configuration Data	Variable	SIPRnet	Collaborative	0.2 Hour	Variable	MIL-STD
	Satellite Access Authorizati	Provide Satellite Access Authorization (Mi	Request Mission IP Address (Network Acce	Data/Text	Satellite Access Authorization	Variable	SIPRnet	One Way	0.1 Hour	Variable	MIL-STD
	Satellite Access Authorizati	Provide Satellite Access Authorization (Mi	USER	Data/Text	Satellite Access Authorization	Variable	SIPRnet	One Way	0.1 Hour	Variable	MIL-STD
5, Network Access Authority to Mission planning	Network Scenarios	Create Network Scenarios (Network Acces	Deconflict GARs (Network Access Authority	Data/Text	Network Scenario Data	Variable	SIPRnet	Collaborative	1 Hour	Variable	MIL-STD
	Gateway Access Request (S	Deconflict GARs (Network Access Authorit	Deconflict SARS (Mission Planning)	Data/Text	Gateway Data	Variable	SIPRnet	Collaborative	0.1 Hour	Variable	MIL-STD
6, Network Access Authority to Network Operations/USER	Gateway Access Authorizati	Develop Gateway Access Authorization (N	Preposition Network Service (Network Ope	Data/Text	Gateway Access Authorization	Variable	SIPRnet	One Way	0.2 Hour	Variable	MIL-STD
	Gateway Access Authorizati	Develop Gateway Access Authorization (N	USER	Data/Text	Gateway Access Authorization	Variable	SIPRnet	One Way	0.2 Hour	Variable	MIL-STD
7, Network Access Authority to Network Operations	Gateway Access Request (S	Deconflict GARs (Network Access Authorit	Request Mission IP Address (Network Acce	Data/Text	Gateway Data	Variable	SIPRnet	One Way	0.1 Hour	Variable	MIL-STD
	Mission IP Request	Request Mission IP Address (Network Acc	Assign Mission IP Address (Network Operat	Data/Text	Mission IP Request Data	Variable	SIPRnet	One Way	0.1 Hour	Variable	MIL-STD
8, Network Operations To Network Access Authority	Mission IP Authorization As	Assign Mission IP Address (Network Opera	Develop Gateway Access Authorization (Ne	Data/Text	Mission IP Data	Variable	SIPRnet	One Way	0.01 Hour	Variable	MIL-STD
8, Network Operations To External	Network Service	Preposition Network Service (Network Op	External	Data/Text	Network Service	Variable	SIPRnet	Collaborative	0.1 Hour	Variable	MIL-STD

Figure 22. DoDAF OV-3: Operational Resource Flow Matrix

Service Area:				
		Technology Forecast		
Service Category	Service Standard	Short (0-1 yr)	Near Term (1-3 yrs)	Long Term (3-5 yrs)
Sensor	Electro Optical	15 Mbps	30 Mbps	50 Mbps

Service Area:				
		Technology Forecast		
Service Category	Service Standard	Short (0-1 yr)	Near Term (1-3 yrs)	Long Term (3-5 yrs)
Sensor	Infrared	8 Mbps	30 Mbps	50 Mbps

Service Area:				
		Technology Forecast		
Service Category	Service Standard	Short (0-1 yr)	Near Term (1-3 yrs)	Long Term (3-5 yrs)
Sensor	Synthetic Aperture Radar	6 Mbps	8 Mbps	10 Mbps

Service Area:				
		Technology Forecast		
Service Category	Service Standard	Short (0-1 yr)	Near Term (1-3 yrs)	Long Term (3-5 yrs)
Comm	RF Link	20 Mbps	83 Mbps	274 Mbps

Figure 23. DoDAF SV-9: Services Technology and Skills Forecast

OV-6a Rules Model: Provide Satellite Access Authorization

1. Conditional Imperative: If mission communications requirements for Satellite Access have been established and are provided, then activity *Plan SATCOM Requirements* has been triggered.
2. Conditional Imperative: If SATCOM Resources are required as determined by the Network Access Authority, then the gateway access request, with the caveat of *SATCOM Resources Required*, must be coordinated through Mission Planning.
    a. If not, then the gateway access request, with the caveat of *SATCOM Resources Not Required* does not need coordination with Mission Planning.
3. Imperative: After the Gateway Access Authorization is developed, it will be provided to the USER, Mission Planning and Network Operations.
4. Imperative: After the Gateway Access Authorization is provided to Network Operations, the Network Service will be prepositioned to make network service available to the USER.

Appendix C Additional Figures and Tables

Table 6. Law and Policy DoDAF Supports

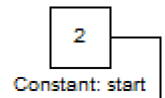
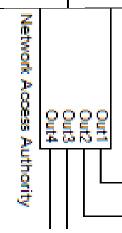
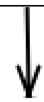
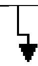
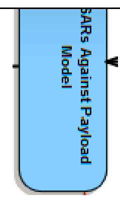
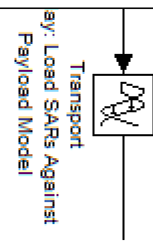
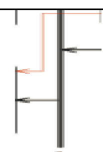
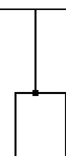
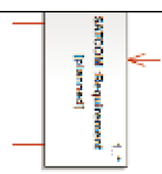
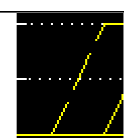
Policy / Guidance	Description
Clinger-Cohen Act of 1996	Recognizes the need for Federal Agencies to improve the way they select and manage IT resources and states, "information technology architecture, with respect to an executive agency, means an integrated framework for evolving or maintaining IT and acquiring new IT to achieve the agency's strategic goals and information resources management goals." Chief Information Officers are assigned the responsibility for "developing, maintaining, and facilitating the implementation of a sound and integrated IT architecture for the executive agency".
E-Government Act of 2002	Calls for the development of Enterprise Architecture to aid in enhancing the management and promotion of electronic government services and processes.
Office of Management and Budget Circular A-130	"Establishes policy for the management of Federal information resources" and calls for the use of Enterprise Architectures to support capital planning and investment control processes. Includes implementation principles and guidelines for creating and maintaining Enterprise Architectures.
OMB Federal Enterprise Architecture Reference Models (FEA RM)	Facilitates cross-agency analysis and the identification of duplicative investments, gaps, and opportunities for collaboration within and across Federal Agencies. Alignment with the reference models ensures that important elements of the FEA are described in a common and consistent way. The DoD Enterprise Architecture Reference Models are aligned with the FEA RM.
OMB Enterprise Architecture Assessment Framework (EAAF)	Serves as the basis for enterprise architecture maturity assessments. Compliance with the EAAF ensures that enterprise architectures are advanced and appropriately developed to improve the performance of information resource management and IT investment decision making.
General Accounting Office Enterprise Architecture Management Maturity Framework (EAMMF)	"Outlines the steps toward achieving a stable and mature process for managing the development, maintenance, and implementation of enterprise architecture." Using the EAMMF allows managers to determine what steps are needed for improving architecture management.

Table 7. DoDAF Meta-model Groups to Viewpoints and DoD Key Processes

Metamodel Data Groups	View Points	DoD Key Processes
	AV, CV, DIV, OV, PV, StdV, SvcV, SV	JCIDS (J), DAS (D), PPBE (P), System Engineering (S), Operations (O), Portfolio Management (IT and Capability) (C)
Performer	CV, OV, PV, StdV, SvcV, SV	J, D, P, S, O, C
Activity	OV	J, O, C
Resource Flow	AV, CV, DIV, OV, PV, StdV	J, S, O
Data and Information	AV, DIV	J, D, P, S, O, C
Capability	CV, PV, SV, SvcV	J, D, P, S, O, C
Services	CV, StdV, SV	P, S, C
Project	AV, CV, PV, SvcV, SV	D, P, S, C
Training/Skill/Education	OV, SV, SvcV, StdV	J, S, O
Goals	CV, PV	J, D, P, O, C
Rules	OV, StdV, SvcV, SV	J, D, S, O
Measures	SvcV, SV	J, D, S, O, C
Location	SvcV, SV	P, S, O

Appendix D   DoDAF Mapping to Simulink

Table 8. Mapping DoDAF Activity Diagram OV-5b to Simulink

UML Activity Diagram (DoDAF OV-5b)		**Simulink Equivalent used in Model**	
Start: initialization (based on precondition?)			Constants, triggers or any source node can be used.
Swim lanes/Partitions Parties involved in the process			Using subsystems as the equivalent to the swim lanes will allow the Simulink model to show the parties/systems involved and follow more closely to the OV-5b format
Transition Supports modeling of control flow			Connectors (line with arrows) will be used Signals/signal flows are represented by the connectors
Action Does something, automatic transition upon its completion Can be an executable code, represented further in sequence diagrams			Currently, signal delays will be used to represent actions; The longer an action takes, the longer the signal delay, where at the end of the signal delay an indication is shown in the signal the action is complete. If the action has a sequence diagram, it may need a subsystem to model it.
Fork One incoming transition, and multiple outgoing parallel transitions and or object flows.			These can be represented by a demux, a signal branch or even a subsystem with one incoming port and two outgoing ports. A simple signal branch will be used.
Object Node An object produced or used by actions. This allows us to model data flows or object flows			Object nodes or data can be represented by signals in the Simulink model. Signals typically have a numeric value in Simulink. A complete action can show a signal having moved from that action (via 0 or 1) to the next.
Join			An AND logical operator

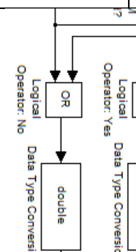
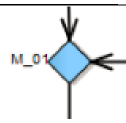
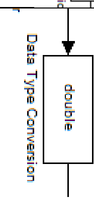
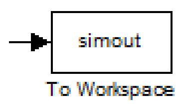
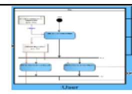
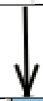
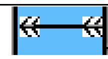
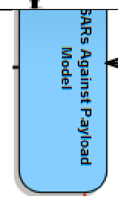
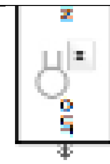
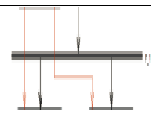
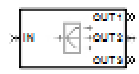
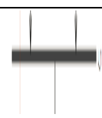
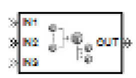
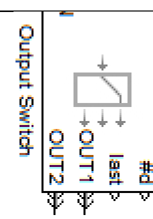
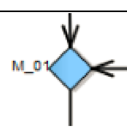
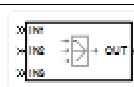
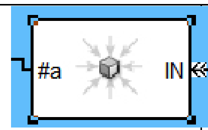
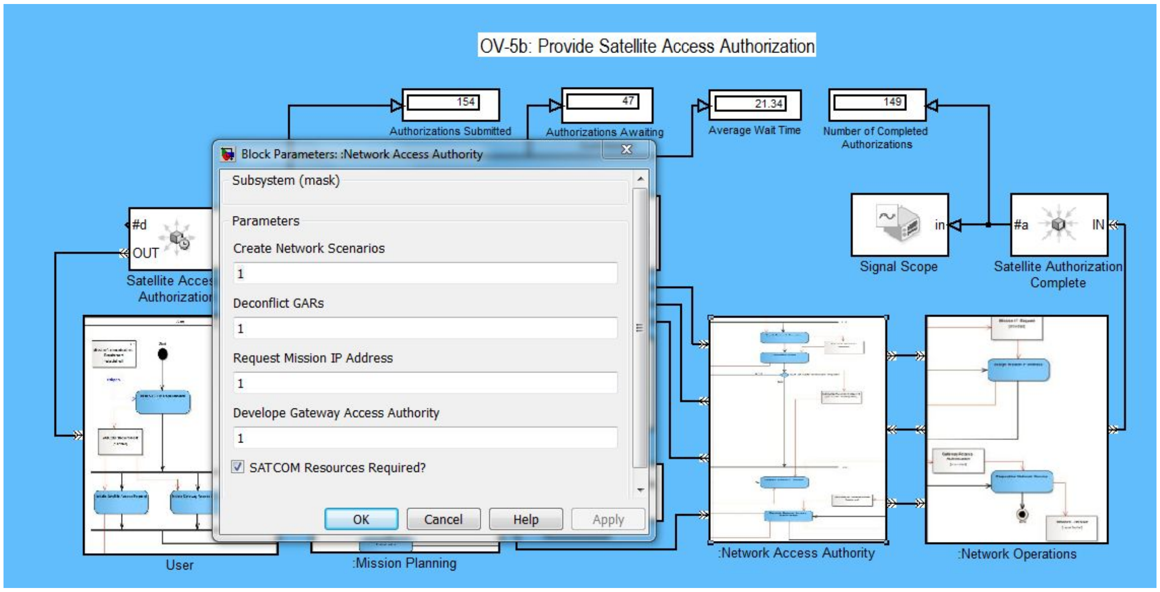
Multiple incoming transitions and/or object flows, on outgoing.			in Simulink serves the same function as an UML join, but has a Boolean output. Using signal delays as actions and a double format signal, requires a converter block to follow the logical operator, converting the signal back into double format.
Outgoing does not happen until ALL the inputs arrive from ALL flows			
Decision			Decisions and merges can be represented by logical operators, or MATLAB Functions. The current method will be using a combination of AND logical operator and an OR logical operator. A *yes* at the decision branch will allow the AND operator to produce a signal, while a *no* won't. The OR operator is used at the merge, because any signal can flow through.
Any branch happens (mutual exclusion)			
If/then/else statements			
Boolean expression			
Provide opportunity for feedback			
Merge			
Any input leads to continuation. This is in contrast to the join			
End: Completion (post condition?)			Assertion, termination, scope or output objects will suffice. For analysis, it is good to have the output objects as the final object as it allows the signal to be output to the desired areas or formats. Simulink models will continue until the last object.

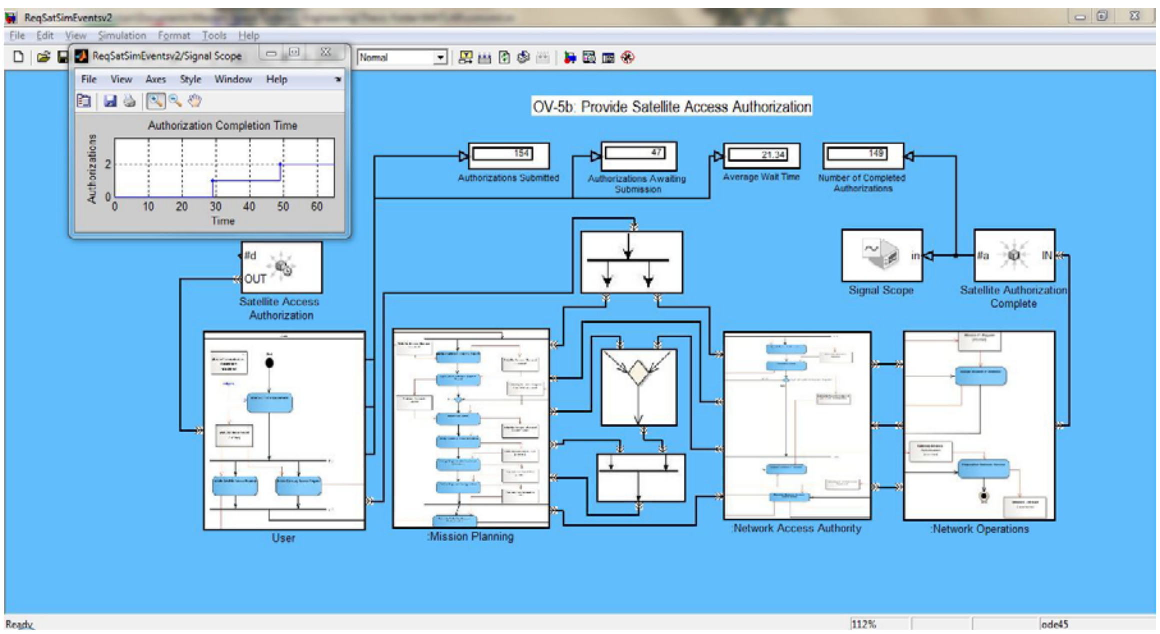
Table 9. Mapping DoDAF Activity Diagram OV-5b to Simulink Toolbox SimEvents

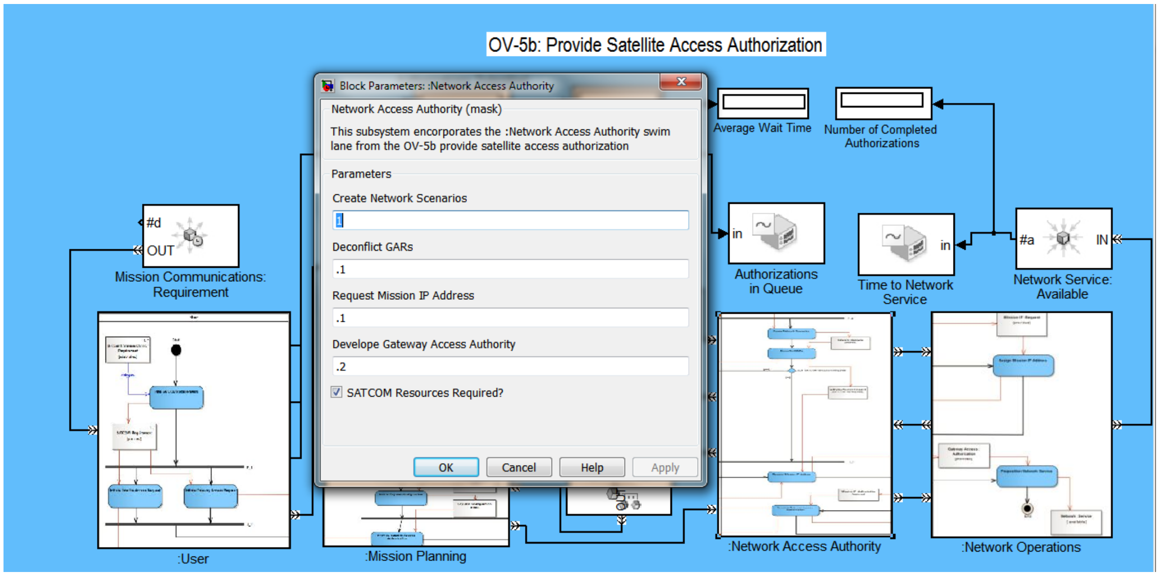
UML Activity Diagram (DoDAF OV-5b)		SimEvents Equivalent	
Start: initialization (based on precondition?)			**Time-Based Entity Generator**
			Generates objects or activities
Swim lanes/Partitions			**Subsystem**
Parties involved in the process			
Transition			**Packet-based transitions**
Supports modeling of control flow			Supports activity flow
Action			**N-Server**
Does something, automatic transition upon its completion			Allows actions to be completed or objects serviced by a specified number of servers
Can be an executable code, represented further in sequence diagrams			Attributes and statistics of servers can be specified in the block
Fork			**Replicate**
One incoming transition, and multiple outgoing parallel transitions and or object flows.			Follows same rule as fork in UML
Object Node			First in First out Queue
An object produced or used by actions. This allows us to model data flows or object flows			Object flow can be visualized from a queue which can output statistics of what objects have processed through it.
Join			**Entity Combiner**
Multiple incoming transitions and/or object flows, one outgoing.			Similar rule as join in UML
Outgoing does not happen until ALL the inputs arrive from ALL flows			Can simulate a join, because outgoing transition does not occur until packets have arrived from all flows
Decision			**Output Switch**
Any branch happens (mutual exclusion)			Output switch determines the output transitions, based on the input in *P*. This can be simulated parameter, or manual decision made real time
If/then/else statements			
Boolean expression			
Provide opportunity for feedback			
Merge			**Path Combiner**
Any input leads to continuation. This is in contrast to the join			Allows all incoming transitions to lead to the single outgoing transition. Any input leads to the continuation, similar to the Merge in UML

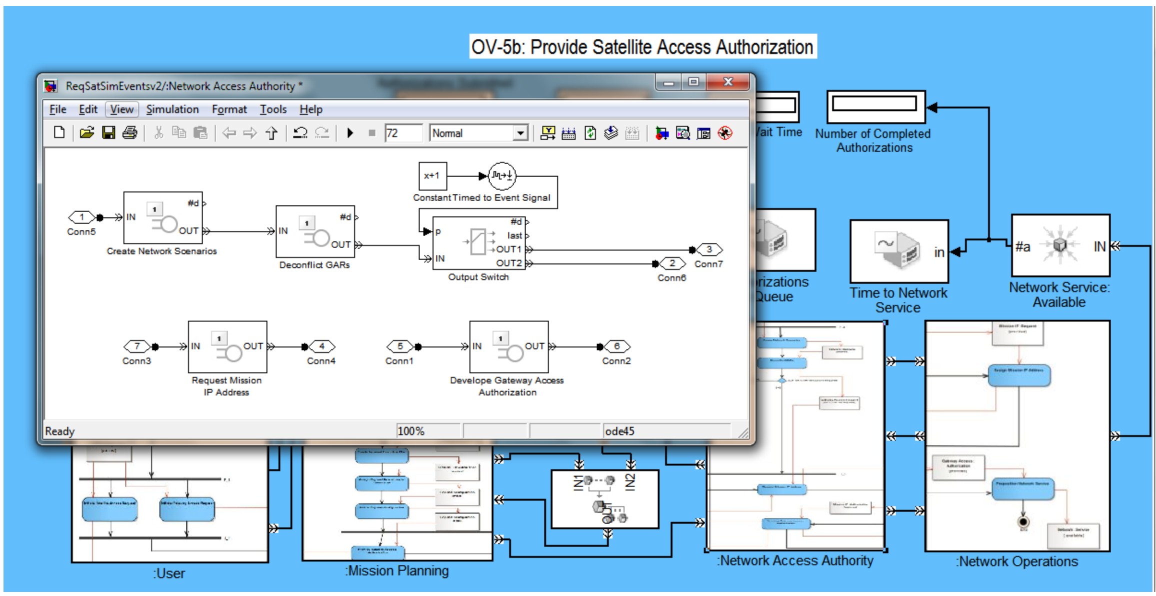
End: Completion			**Entity Sink**
			Ends the activities, and allows for output statistics

Appendix E   Screenshots of OV-5b Executable Architecture






OV-5b: Provide Satellite Access Authorization
Block Parameters: :Network Access Authority
Network Access Authority (mask)
This subsystem encorporates the :Network Access Authority swim lane from the OV-5b provide satellite access authorization
Parameters
Create Network Scenarios
Deconflict GARs
.1
Request Mission IP Address
.1
Develope Gateway Access Authority
.2
SATCOM Resources Required?
OK
Cancel
Help
Apply
Average Wait Time
Number of Completed Authorizations
Mission Communications: Requirement
Authorizations in Queue
Time to Network Service
Network Service: Available
:User
:Mission Planning
:Network Access Authority
:Network Operations


OV-5b: Provide Satellite Access Authorization
ReqSatSimEventsv2/:Network Access Authority *
File Edit View Simulation Format Tools Help
Constant Timed to Event Signal
Create Network Scenarios
Deconflict GARs
Output Switch
Conn5
Conn6
Conn7
Conn3
Conn4
Conn1
Conn2
Request Mission IP Address
Develope Gateway Access Authorization
Ready
100%
ode45
Number of Completed Authorizations
Time to Network Service
Network Service: Available
:User
:Mission Planning
:Network Access Authority
:Network Operations

Appendix F Further Questionnaire Results Analysis

Q1	**Do you have any prior experience or are you familiar with Simulink/MATLAB**					Familiar with MATLAB				Familiar with DoDAF				Familiar with the RPA Systems				Familiar with All 3			
Code	Value	Frequency	Percent	Total	10	Frequency	Percent	Total	6	Frequency	Percent	Total	8	Frequenc	Percent	Total	9	Frequenc	Percent	Total	4
1	No Experience	4	40.00%			0	0.00%			3	37.50%			4	44.44%			0	0.00%		
2	Some Experience	5	50.00%			5	83.33%			4	50.00%			4	44.44%			3	75.00%		
3	Experienced	1	10.00%			1	16.67%			1	12.50%			1	11.11%			1	25.00%		
Q2	**With DoDAF?**					Familiar with MATLAB				Familiar with DoDAF				Familiar with the RPA Systems				Familiar with All 3			
Code	Value	Frequency	Percent		10	Frequency	Percent	Total	6	Frequency	Percent	Total	8	Frequenc	Percent	Total	9	Frequenc	Percent	Total	4
1	No Experience	2	20.00%			1	16.67%			0	0.00%			1	11.11%			0	0.00%		
2	Some Experience	3	30.00%			2	33.33%			3	37.50%			3	33.33%			1	25.00%		
3	Experienced	5	50.00%			3	50.00%			5	62.50%			5	55.56%			3	75.00%		
Q3	**With the RPA Communications Systems presented in the architectural products?**					Familiar with MATLAB				Familiar with DoDAF				Familiar with the RPA Systems				Familiar with All 3			
Code	Value	Frequency	Percent	Total	10	Frequency	Percent	Total	6	Frequency	Percent	Total	8	Frequenc	Percent	Total	9	Frequenc	Percent	Total	4
1	No Experience	2	20.00%			2	33.33%			1	12.50%			1	11.11%			0	0.00%		
2	Some Experience	3	30.00%			2	33.33%			3	37.50%			3	33.33%			2	50.00%		
3	Experienced	5	50.00%			2	33.33%			4	50.00%			5	55.56%			2	50.00%		
Q4	**Based on the samples and demo provided could the systems architecture be effectively evaluated in an executable**					Familiar with MATLAB				Familiar with DoDAF				Familiar with the RPA Systems				Familiar with All 3			
Code	Value	Frequency	Percent	Total	10	Frequency	Percent	Total	6	Frequency	Percent	Total	8	Frequenc	Percent	Total	9	Frequenc	Percent	Total	4
1	Completely Ineffe	0	0.00%			0	0.00%			0	0.00%			0	0.00%			0	0.00%		
2	Somewhat Effecti	3	30.00%			2	33.33%			3	37.50%			3	33.33%			1	25.00%		
3	Largely Effective	5	50.00%			3	50.00%			3	37.50%			4	44.44%			2	50.00%		
4	Completely Effect	2	20.00%			1	16.67%			2	25.00%			2	22.22%			1	25.00%		
Q5	**Is the executable architecture effective for allowing a dynamic analysis of the systems architecture it represents?**					Familiar with MATLAB				Familiar with DoDAF				Familiar with the RPA Systems				Familiar with All 3			
Code	Value	Frequency	Percent	Total	9	Frequency	Percent	Total	6	Frequency	Percent	Total	7	Frequenc	Percent	Total	8	Frequenc	Percent	Total	4
1	Completely Ineffe	0	0.00%			0	0.00%			0	0.00%			0	0.00%			0	0.00%		
2	Somewhat Effecti	3	33.33%			3	50.00%			2	28.57%			2	25.00%			1	25.00%		
3	Largely Effective	4	44.44%			2	33.33%			3	42.86%			3	37.50%			2	50.00%		
4	Completely Effect	3	33.33%			1	16.67%			2	28.57%			3	37.50%			1	25.00%		
Q6	**Has Simulink been effectively used to convert the DoDAF Architectural Products to an executable format?**					Familiar with MATLAB				Familiar with DoDAF				Familiar with the RPA Systems				Familiar with All 3			
Code	Value	Frequency	Percent	Total	10	Frequency	Percent	Total	6	Frequency	Percent	Total	8	Frequenc	Percent	Total	9	Frequenc	Percent	Total	4
1	Completely Ineffe	0	0.00%			0	0.00%			0	0.00%			0	0.00%			0	0.00%		
2	Somewhat Effecti	2	20.00%			2	33.33%			2	25.00%			2	22.22%			1	25.00%		
3	Largely Effective	4	40.00%			2	33.33%			4	50.00%			4	44.44%			2	50.00%		
4	Completely Effect	4	40.00%			2	33.33%			2	25.00%			3	33.33%			1	25.00%		

Q7	**To analyze architectures, would you consider using Simulink as a tool?**					Familiar with MATLAB				Familiar with DoDAF				Familiar with the RPA Systems				Familiar with All 3			
Code	Value	Frequency	Percent	Total	10	Frequency	Percent	Total	6	Frequency	Percent	Total	8	Frequency	Percent	Total	9	Frequency	Percent	Total	4
1	Won't Consider	1	10.00%			0	0.00%			1	12.50%			1	11.11%			0	0.00%		
2	Maybe Consider	3	30.00%			2	33.33%			2	25.00%			3	33.33%			1	25.00%		
3	Will Consider	6	60.00%			4	66.67%			5	62.50%			5	55.56%			3	75.00%		
Q8	**evaluating the Systems or System of Systems architecture as described by DoDAF products?**					Familiar with MATLAB				Familiar with DoDAF				Familiar with the RPA Systems				Familiar with All 3			
Code	Value	Frequency	Percent			Frequency	Percent	Total	6	Frequency	Percent	Total	8	Frequency	Percent	Total	9	Frequency	Percent	Total	4
1	Completely Ineffe	0	0.00%	Total	10	0	0.00%			0	0.00%			0	0.00%			0	0.00%		
2	Somewhat Effectiv	4	40.00%			3	50.00%			4	50.00%			4	44.44%			2	50.00%		
3	Largely Effective	4	40.00%			2	33.33%			3	37.50%			4	44.44%			2	50.00%		
4	Completely Effect	2	20.00%			1	16.67%			1	12.50%			1	11.11%			0	0.00%		
Q9	**As presented in Simulink, does this executable architecture effectively represent the DoDAF architectural products?**					Familiar with MATLAB				Familiar with DoDAF				Familiar with the RPA Systems				Familiar with All 3			
Code	Value	Frequency	Percent	Total	10	Frequency	Percent	Total	6	Frequency	Percent	Total	8	Frequency	Percent	Total	9	Frequency	Percent	Total	4
1	Completely Ineffe	0	0.00%			0	0.00%			0	0.00%			0	0.00%			0	0.00%		
2	Somewhat Effectiv	4	40.00%			4	66.67%			4	50.00%			4	44.44%			3	75.00%		
3	Largely Effective	4	40.00%			1	16.67%			3	37.50%			4	44.44%			1	25.00%		
4	Completely Effect	2	20.00%			1	16.67%			1	12.50%			1	11.11%			0	0.00%		
Q10	**Do the Simulink executable models present an accurate depiction of the DoDAF architectural products just as UML**					Familiar with MATLAB				Familiar with DoDAF				Familiar with the RPA Systems				Familiar with All 3			
Code	Value	Frequency	Percent	Total	6	Frequency	Percent	Total	5	Frequency	Percent	Total	5	Frequency	Percent	Total	5	Frequency	Percent	Total	4
1	Entirely Innacurate	0	0.00%			0	0.00%			0	0.00%			0	0.00%			0	0.00%		
2	Somewhat Accurat	2	33.33%			2	40.00%			2	40.00%			2	40.00%			2	50.00%		
3	Largely Accurate	3	50.00%			2	40.00%			2	40.00%			2	40.00%			2	50.00%		
4	Completely Accura	2	33.33%			1	20.00%			1	20.00%			1	20.00%			0	0.00%		
Q11	**Is Simulink/MATLAB an effective product for analyzing architectures?**					Familiar with MATLAB				Familiar with DoDAF				Familiar with the RPA Systems				Familiar with All 3			
Code	Value	Frequency	Percent	Total	8	Frequency	Percent	Total	4	Frequency	Percent	Total	7	Frequency	Percent	Total	8	Frequency	Percent	Total	3
1	Completely Ineffe	0	0.00%			0	0.00%			0	0.00%			0	0.00%			0	0.00%		
2	Somewhat Effectiv	4	50.00%			3	75.00%			4	57.14%			4	50.00%			2	66.67%		
3	Largely Effective	3	37.50%			1	25.00%			2	28.57%			3	37.50%			1	33.33%		
4	Completely Effect	1	12.50%			0	0.00%			1	14.29%			1	12.50%			0	0.00%		
Q12	**Given your knowledge, the samples and demo provided, would you consider utilizing executable architecting?**					Familiar with MATLAB				Familiar with DoDAF				Familiar with the RPA Systems				Familiar with All 3			
Code	Value	Frequency	Percent	Total	8	Frequency	Percent	Total	5	Frequency	Percent	Total	6	Frequency	Percent	Total	7	Frequency	Percent	Total	3
1	Won't Consider	0	0.00%			0	0.00%			0	0.00%			0	0.00%			0	0.00%		
2	Maybe Consider	1	12.50%			1	20.00%			1	16.67%			1	14.29%			0	0.00%		
3	Will Consider	7	87.50%			4	80.00%			5	83.33%			6	85.71%			3	100.00%		

Bibliography

AbuSharekh, A., Kansal, S., Zaidi, A. K., & Levis, A. H. (2007). Modeling Time in DoDAF Compliant Executable Architectures.

Beal, R. J., Hendrix, J. P., McMurray, G. P., & Stewart, W. C. (2005). *Executable Architectures and Their Application to a Geographically Distributed Air Operations Center.* Air Force Institute of Technology.

Bornejko, T. L., Glasscock, C. G., & Sprenkle, D. R. (2008). *Creating A Discrete Event Simulation to Determine the Military Worth of Developing an Electronic Warfare Battle Manager Function Within an Airborne Electronic Attack Systems Architecture .*

Department of Defense. (2012). *DoDAF Architecture Framework Version 2.02.* Retrieved from http://cio-nii.defense.gov/sites/dodaf20/index.html

Dietrichs, T., Griffin, R., Schuettke, A., & Slocum, M. (2006). *Integrated Architecture Study for Weapon Borne Battle Damage Assessment System Evaluation.*

Eller, J., Hazel, B., & Rooney, B. (2008). *GLOBAL PERSISTENT ATTACK: A SYSTEMS ARCHITECTURE, PROCESS MODELING, AND RISK ANALYSIS APPROACH.*

Garcia, J. (2007). *Executable Architecture Analysis Modeling for Network Testing and Evaluation in an HLA and TENA environment.* SimIS, Inc.

Griendling, K., & Marvis, D. (2011). Development of a DoDAF-Based Executable Architecting Approach to Analyze System of System Alternatives.

K. Jensen, L. K. (2009). *Coloured Petri Nets.* Berlin Heidelber: Springer-Verlag .

Levis, A. H., & Wagenhals, L. W. (2000). *C4ISR architectures: I. Developing a process for C4ISR architecture design.* John Wiley & Sons, Inc.

MathWorks. (2012). *Simulink*. Retrieved 2012, from http://www.mathworks.com/products/simulink/

Mittal, S. (2006). Extending DoDAF to Allow Integrated DEVS-Based Modeling and Simulation.

Mozaffari, M., Harounabadi, A., & Mirabedini, S. (2011). A Method for Validating the Behavior of Enterprise Architecture. *World Applied Sciences Journal*.

Rechtin, M. W. (2009). *The Art of Systems Architecting.*

Sjostedt, C.-J. (n.d.). Mapping Simulink to UML in the design of embedded systems: Investigating scenarios and transformations.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
01-09-2012	Master's Thesis	04 October 2012 – 01 September 2012

4. TITLE AND SUBTITLE
A STUDY OF EXECUTABLE MODEL BASED SYSTEMS ENGINEERING FROM DODAF USING SIMULINK

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Hanoka, Weston J., First Lieutenant, USAF
Ryan, Michael H., First Lieutenant, USAF

5d. PROJECT NUMBER
AFIT/GSE/ENV/12-S05DL

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)
Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/ENV)
2950 Hobson Way, Building 640
WPAFB OH 45433-8865

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GSE/ENV/12-S05DL

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Lt Col Mark Brykowytch, Advanced Concepts Division, Military Satellite Communications Division, Space and Missile Systems Center (SMC/MCX)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT
Diagrams and visuals often cannot adequately capture a complex system's architecture for analysis. The Department of Defense Architectural Framework (DoDAF), written to follow the Unified Modeling Language (UML), is a collection of mandated common architectural products for interoperability among the DoD components. In this study, DoDAF products from as-is Remotely Piloted Aircraft (RPA) Satellite Communication (SATCOM) systems have been utilized for the creation of executable architectures as part of an Executable Model Based Systems Engineering (EMBSE) process. EMBSE was achieved using Simulink, a software tool for modeling, simulating and analyzing dynamic systems. This study has demonstrated that DoDAF products can be created and executed following the rules of UML for analysis. It has also shown that DoDAF products can be utilized to build analysis models. Furthermore, these analysis models and executable architectures have been presented to a panel of experts on the topic. The comments and study results show a desire for executable architectures as well as their viability as presented in Simulink. This study concludes there is a need, a use and a method to implement objective analysis using EMBSE from DoDAF products in Simulink for current and future DoD systems.

15. SUBJECT TERMS
DoDAF, Executable Architecture, EMBSE, Systems Engineering, Simulink, UML

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
					Lt Col Brent Langhals
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)*
U	U	U	U	95	(937) 255-6565, x 4352 (Brent.Langhals@afit.edu)