NAVAL
POSTGRADUATE
SCHOOL

MONTEREY, CALIFORNIA

THESIS

EXPLORING THE CAPABILITIES OF CAMEO ENTERPRISE ARCHITECTURE TO INTEGRATE AND OPERATE A DYNAMIC MODEL OF MK 15 PHALANX CIWS

by

Alexander X. Yeiser

June 2021

Thesis Advisor:	Oleg A. Yakimenko
Co-Advisor:	Saulius Pavalkis (3DS)
Second Reader:	Fotis A. Papoulias

REPORT DOCUMENTATION PAGE	*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY ***(Leave blank)***	**2. REPORT DATE** June 2021	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE EXPLORING THE CAPABILITIES OF CAMEO ENTERPRISE ARCHITECTURE TO INTEGRATE AND OPERATE A DYNAMIC MODEL OF MK 15 PHALANX CIWS	**5. FUNDING NUMBERS**
6. AUTHOR(S) Alexander X. Yeiser	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

An increase in technology and software capabilities has led to an increase in computer modeling to better understand systems. There are many methods of modeling and various software programs created for this purpose. By exploring a program called Cameo Enterprise Architecture (Cameo), made by the software company No Magic Incorporated, the goal is to utilize this state-of-the-art software, used for model-based systems engineering (MBSE) by the DOD research labs, to drive a standalone physical model of the weapon system. Cameo provides a variety of tools for in-depth system modeling and architecture, and the goal of this thesis is to explore different applications of Cameo and techniques involved in using it to model real systems. This thesis presents a physical model of the Mk 15 Phalanx Close-in Weapon System (CIWS) built using the Lego Mindstorms kit and then develops a routine to control it from Cameo. Such an integrated model is thought of as a useful tool to verify the requirements of the system at the early stages of its design. The developed integral model is tested using realistic scenarios to determine the effectiveness of the diagrams and models created in Cameo.

14. SUBJECT TERMS MBSE, Cameo	**15. NUMBER OF PAGES** 63
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

EXPLORING THE CAPABILITIES OF CAMEO ENTERPRISE ARCHITECTURE TO INTEGRATE AND OPERATE A DYNAMIC MODEL OF MK 15 PHALANX CIWS

Alexander X. Yeiser
Ensign, United States Navy
BS, U.S. Naval Academy, 2020

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
June 2021

Approved by: Oleg A. Yakimenko
Advisor

Saulius Pavalkis
Co-Advisor

Fotis A. Papoulias
Second Reader

Ronald E. Giachetti
Chair, Department of Systems Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

An increase in technology and software capabilities has led to an increase in computer modeling to better understand systems. There are many methods of modeling and various software programs created for this purpose. By exploring a program called Cameo Enterprise Architecture (Cameo), made by the software company No Magic Incorporated, the goal is to utilize this state-of-the-art software, used for model-based systems engineering (MBSE) by the DOD research labs, to drive a standalone physical model of the weapon system. Cameo provides a variety of tools for in-depth system modeling and architecture, and the goal of this thesis is to explore different applications of Cameo and techniques involved in using it to model real systems. This thesis presents a physical model of the Mk 15 Phalanx Close-in Weapon System (CIWS) built using the Lego Mindstorms kit and then develops a routine to control it from Cameo. Such an integrated model is thought of as a useful tool to verify the requirements of the system at the early stages of its design. The developed integral model is tested using realistic scenarios to determine the effectiveness of the diagrams and models created in Cameo.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

Effective modeling and integration are important capabilities critical to a systems engineer. This involves breaking down complex systems into simpler parts and presenting them in a way that is less complicated to understand. To perform such tasks, systems engineers can use advanced software tools and standardized modeling methods. Cameo is a program that allows for systems architecture, as well as more in-depth modeling and simulation. The goal of this thesis is to explore the capabilities of Cameo to model a weapon system deployed by the U.S. Navy.

Specifically, different aspects of the MK 15 Phalanx CIWS can be represented in Cameo. The model is created using several different diagrams commonly used in the Systems Modeling Language (SysML). These diagrams include: block definition diagrams, internal block diagrams, state machine diagrams, and activity diagrams. Individually, each of these diagrams is used to model systems differently and they are effective representations of the system. When the diagrams are used together, along with Cameo simulation software, they can be made executable and used for simulation purposes. While Cameo provides a digital model of the weapon system, a Lego Mindstorms EV3 development kit can be used to create a physical model. Lego infrared sensors can receive inputs and send the data to the model made in Cameo. The model reacts based on changing inputs and sends control signals to motors built into the model.

The integration between the Lego Mindstorms and Cameo can be done successfully, but it comes with certain limitations. There is a library that was created for Cameo that is used to make the connections between the model and the Lego hardware which allows for the model to retrieve data and send command signals. However, this library was not created to access the full capabilities of the Lego hardware, such as changing the sensor modes and retrieving inputs from the motor. The limitations did not allow for in-depth testing of the model, as intended, but still permitted effective integration between Cameo and Lego, which is the overarching goal of the project.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGEMENTS

I would like to recognize and thank Dr. Saulius Pavalkis for introducing me to Cameo Enterprise Architecture and for helping to troubleshoot any issues that arose along the way. When there was no apparent answer in sight, he seemed to provide a quick solution.

I would also like to thank Professor Papoulias for taking the time to be my second reader and for being a part of this research process.

I am extremely grateful for my research advisor, Professor Oleg Yakimenko, who guided me along the way. At the same time, he allowed me to work at my own pace and take on this research with authority and commitment.

Finally, I would like to thank my family and friends for pushing me to do the best I can throughout this research. They motivate me to always explore new ideas and pursue all my life goals.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

This chapter introduces the problem this thesis deals with. First, it presents the current trend in Systems Engineering and introduces one of the most advanced model-based system engineering tools, chosen to experiment within this study. Next, it describes one of the particular DOD-related problems that is addressed in this study and formulates a research problem. The section ends with an outline of the thesis.

A. BACKGROUND

Understanding how to model system requirements and functions while utilizing complex software tools is a critical part of systems engineering. Cameo Enterprise Architecture (Cameo), made by the software company No Magic Incorporated, is state-of-the-art model-based systems engineering (MBSE) tool and utilizes various modeling languages and diagram capabilities to create highly sophisticated systems architectures. Other modeling tools can accomplish similar goals, but they may require less modeling experience and will not perform at as high a level. Given the various less complicated and user-friendly modeling tools, systems engineers must understand the limitations of these tools and learn to work with more complex tools. Cameo creates the opportunity for systems engineers to explore higher-fidelity software when modeling different aspects of a system.

It is essential that systems engineers not only understand this type of software but know how to integrate it with real systems or their digital twins. Digital twin refers to a computer model of a system that can be used to represent the actual system when performing developmental or operational tests. This saves both time and money because the real system does not need to be used. If a digital twin is not yet available for a system, the conceptual design stage may rely on a system mockup or prototype. In this case, it is important to be able to conduct conceptual design within the same development environment without switching back and forth between MBSE tool and software communicating with the prototype.

For example, consider Mk 15 Phalanx Close-in Weapon System (CIWS) shown in Figure 1, “a fast-reaction, rapid-fire 20-millimeter gun system that provides U.S. Navy ships with a terminal defense against anti-ship missiles that have penetrated other fleet defenses” (Pike 2003). Suppose there is a need to introduce some modifications to this existing system and in lieu of a high-fidelity computer model, a mockup is used. This mockup would be built to allow reproducing some functionalizes of the future system and therefore could be extremely useful in exploring the design space.

Figure 1. Mk 15 Phalanx CIWS with Component Labels. Source: Pike (2003).

The question is then, whether any MBSE tool and Cameo, in particular, would allow explicitly communicating with this mockup, passing on varied design parameters, simulating a mission, and reading back the values affecting the design decision or providing material for parametric (trade-off) studies/optimization. This is what this thesis is attempting to answer.

B. RESEARCH OBJECTIVES

The overall goal of this research is to explore Cameo Enterprise Architecture and its applicability to control a prototype of some system. Specifically, this research employs a physical model of a CIWS made using Lego Mindstorms. Cameo is used to describe the system and its requirements and then to physically control the model. This not only requires modeling the system using a software tool but also integrating Cameo and its computer models with the Lego Mindstorms to control the system, both of which are important functions that a systems engineer would benefit from. The integrated Cameo/Mk 15 setup is supposed to receive different inputs using Lego Mindstorms sensors, and the models built in Cameo use these inputs to output certain control signals. These control signals move the position of the physical turret model based on varying inputs. The results from the models will be used to determine whether Cameo was able to successfully actuate the model to meet the system requirements.

The specific research questions this thesis addresses are:

- Whether Cameo Enterprise Architecture can be used to model and control systems created using Lego Mindstorm kits.
- Whether the integrated Cameo/Lego Mindstorm tool is useful to conduct parametric and trade-off studies and verify system-level requirements?

C. BENEFIT OF STUDY

Being able to model systems in their entirety is a critical aspect of systems engineering. Systems Engineers use models and systems architecture to connect stakeholder requirements to functional aspects of a system, to integrate systems, and to ensure there is a general understanding of the system. This thesis explores ways to apply these aspects of systems engineering to high-performance software and creates opportunities for future students to further explore the same software and all of its capabilities. Students of the Naval Postgraduate School within the systems engineering curriculum will be able to move on from other less complicated software and begin to work with a complex modeling tool, with many different applications, that can be used to operate model systems.

In addition to the several practical applications for systems engineers, knowledge and understanding of complex modeling tools can be used throughout the Department of Defense (DOD). Given the numerous weapon systems the DOD employs, it is sensible that the DOD requires advanced software to model these systems. Regardless of whether or not the DOD uses Cameo specifically for its modeling needs, it is important to have adequate skills when it comes to undertaking state-of-the-art modeling software and exploring its capabilities while performing duties for the DOD.

D. THESIS ORGANIZATION

The remainder of the thesis is organized as follow:

- Chapter II introduces systems engineering concepts and their relationship to Model-based systems engineering, as well as the recent effort from the DOD to integrate digital engineering into their acquisition processes. This chapter also discusses different modeling languages and gives insight into the reason Cameo stands out amongst other modeling tools.
- Chapter III provides background on the CIWS and how it operates in the fleet. The Lego Mindstorms model is displayed in this chapter as well as the process of how the connection was made between Cameo and the model. Other physical connections and their respective inputs and outputs are presented as well.
- Chapter IV presents the various diagrams that were created to model the system. The different diagrams give a broad overview of the system, a view of what components are needed and how they are connected, and a look at how the model receives input and then chooses control signals. The outcome of the simulated model is also discussed.
- Chapter V summarizes the work that took place throughout this research and discusses whether or not the research questions were adequately addressed. Further research opportunities and future applications are also explored.

II. MBSE AND CAMEO

This chapter introduces and defines systems engineering, and more specifically, model-based systems engineering (MBSE). First, concepts of digital engineering are broken down and discussed. Second, information is given on why the DOD is pushing to incorporate MBSE into its acquisition process. This involves taking a look at their digital engineering strategy and its major goals. Next, this chapter describes the different modeling languages and discusses what is unique about SysML.

A. DIGITAL ENGINEERING

Lately, the DOD has shown a need for an improvement in the acquisition process. Traditional acquisition processes are document-intensive and often lead to cost and time overruns. In June of 2018, the Under Secretary of Defense for Research and Developent released the U.S. DOD Digital Engineering Strategy. This strategy describes five goals to streamline DOD acquisition process through digital engineering, of which Model-based systems engineering (MBSE) is a subset (Uppal 2021). The DOD's efforts to adopt digital engineering into their acquisition process demonstrate a need for in-depth research and testing to effectively standardize the use of MBSE throughout the system development process.

Systems engineering encompasses a wide variety of engineering components and there are many different accepted definitions. As defined by the International Counsel on Systems Engineering (INCOSE), it is "an interdisciplinary approach and means to enable the realization of successful systems" (INCOSE 2021). Systems engineers are tasked with maintaining a broad view of a system, as a "major characteristic of the engineering of systems is the attention devoted to the entire life cycle of the system" (Buede 2009, 3). In general, there is a great deal of information that systems engineers need to utilize, organize, and maintain such as stakeholder requirements, system functions, interfaces, and important documentation, especially given the increase in the complexity of modern-day systems.

Effective system modeling using modern technology and software gives systems engineers a wide variety of tools and methods to assist in the process. Model-based systems engineering (MBSE) is a "formalized application of modeling techniques to the definition, analysis, design, verification, and validation of a system throughout its life cycle from inception to use" (Pavalkis and Wang 2019,1). MBSE has become an integral part of the systems engineering process, as it is used at many different points in a systems life cycle. Therefore, systems engineers need to have a strong understanding of how to properly approach and apply MBSE, which can be difficult given problems that may arise throughout the process.

Systems engineering demands being able to understand a problem and approach it in an organized and methodical manner. This can involve being able to properly integrate different systems or software that was not specifically meant to be used together. A paper by Dr. Dirk Zwemer from InterCAX LLC describes an attempt to use software called ParaMagic and SysML with UPDM (Unified Profile for DoDaf/MoDAF). SysML and UPDM have a complicated relationship, in that UPDM is a variant of SysML used for defense purposes. Despite one being a variant of the other, the objectives of their model constructions are quite different. The project used a real UAV system to create scenarios that could be simulated by the combined software, and the results were a general success, with a few lessons learned.

The goal of this thesis is to address the concept of complex integration and several aspects of the example from Dr. Zwemer above. The Lego Mindstorm kit to be used in this project was designed to operate with a user-friendly desktop software created by Lego. As useful as this may be, this type of software has limitations and may not be able to support all the software tools, such as diagrams and simulations, that systems engineering requires. Integrating the Lego Mindstorm hardware with Cameo Enterprise Architecture software exemplifies that ability to incorporate components that were not specifically meant to work together, while at the same time providing an example for future systems engineers to follow when it comes to working with Cameo, or other sophisticated software.

B. GOVERNMENT PUSH FOR MBSE

The problems the DOD had been facing for a long time came from traditional acquisition processes that over time became normalized. These processes were typically "document-intensive and stove-piped, leading to extended cycle times with systems that are cumbersome to change and sustain" (Office of the Deputy Assistant Secretary of Defense for Systems Engineering 2018). Given the importance of acquisition within the DOD, these extended cycle times could create further problems that put both our country and members of our military at risk. Clearly, there was a need for change, especially since it took the DOD until 2018 to realize that advancements in technology could benefit government acquisition. Figure 2 describes the potential benefits the DOD expects to gain from the transition to digital engineering.

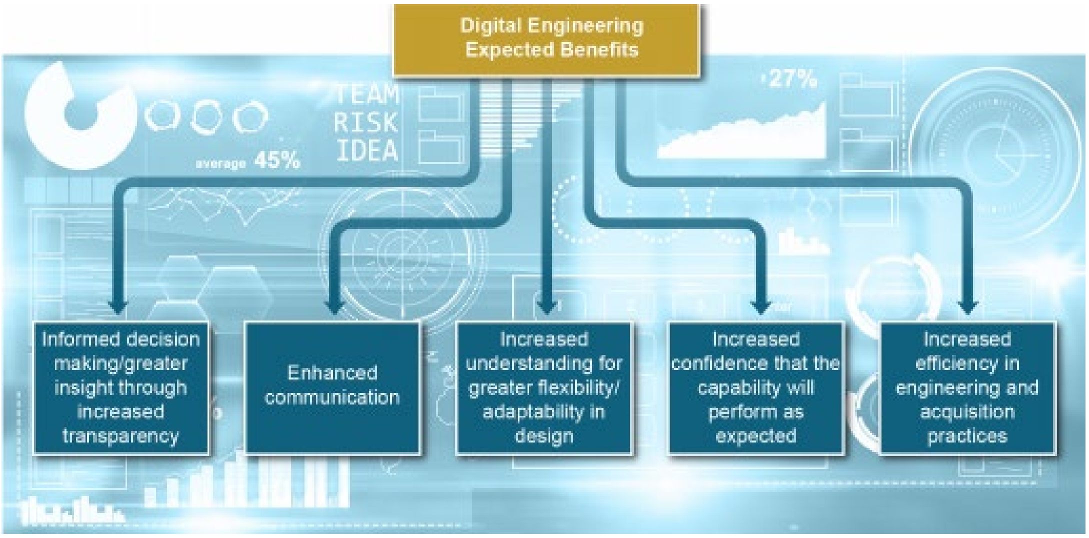


Figure 2. DOD Expected Benefits from Implementing Digital Engineering. Source: Office of the Deputy Assistant Secretary of Defense for Systems Engineering (2018)

The way to obtain these benefits was through the DOD engineering strategy, which is comprised of the following five goals:

1. Formalize the development, integration, and use of models to inform enterprise and program decision-making.
2. Provide an enduring, authoritative source of truth.
3. Incorporate technological innovation to improve the engineering practice.
4. Establish a supporting infrastructure and environments to perform activities, collaborate, and communicate across stakeholders.

5. Transform the culture and workforce to adopt and support digital engineering across the life cycle. (Office of the Deputy Assistant Secretary of Defense for Systems Engineering 2018)

Implementation of these goals across the DOD will take coordinated efforts by all of its components, but will ideally result in a more efficient and effective acquisition process. Since the institution of this new strategy, the DOD has already seen success with the development of the sixth generation strike fighter, the F-35. In September of 2020, Dr. Roper, assistant secretary of the Air Force for acquisition, technology, and logistics, "heralded the need for acquisition agility through digital engineering to disrupt the nation's adversaries" (Wilson 2020). The effort to incorporate MBSE into government acquisition will undoubtedly continue to see success and is being noticed by other organizations outside of the DOD. According to an article from the National Aeronautics and Space Administration (NASA), "the NASA MBSE pathfinder is demonstrating the application of MBSE in space-related systems" (Solberg 2020). As this process continues to become prominent among high-level organizations, the need for systems engineers who are skilled in using modeling tools will increase.

C. UNDERSTANDING MODELING LANGUAGES AND SYSML

According to an Insight article," a model-based systems engineering approach must support collaboration and interoperability at several levels: across global organizations, between disciplines involved in the systems development effort, among design teams within a given discipline, between design and analysis efforts, and between development and manufacturing" (Peak et al. 2009, 40). The developments of modeling languages have helped work towards more interoperability within MBSE. Cameo is capable of utilizing several different languages, but the focus of this project is on the systems modeling language (SysML), which was developed by Object Modeling Group and INCOSE and "is used to model all aspects of a system either directly or through an interface with another model" (Kaslow et al. 2014,2). SysML was not necessarily created from scratch, but rather used many aspects of the Unified Modeling Language (UML) to produce a more specified and standard language for systems engineers. Specifically, SysML "reuses a subset of UML2(UML4SysML), and defines its own extensions" (Guillaume 2010, 3). SysML also has “nine diagrams instead of thirteen diagrams from UML2, making it a smaller language that is easier to learn and apply” (Guillaume 2010, 3). Since its creation, SysML has become widely used in the systems engineering domain and other MBSE tools, such as Innoslate, are capable of supporting SysML.

As discussed, SysML consist of a few UML2 diagrams, as well as additional diagrams unique to SysML. Figure 3 displays the different diagrams that makeup SysML.

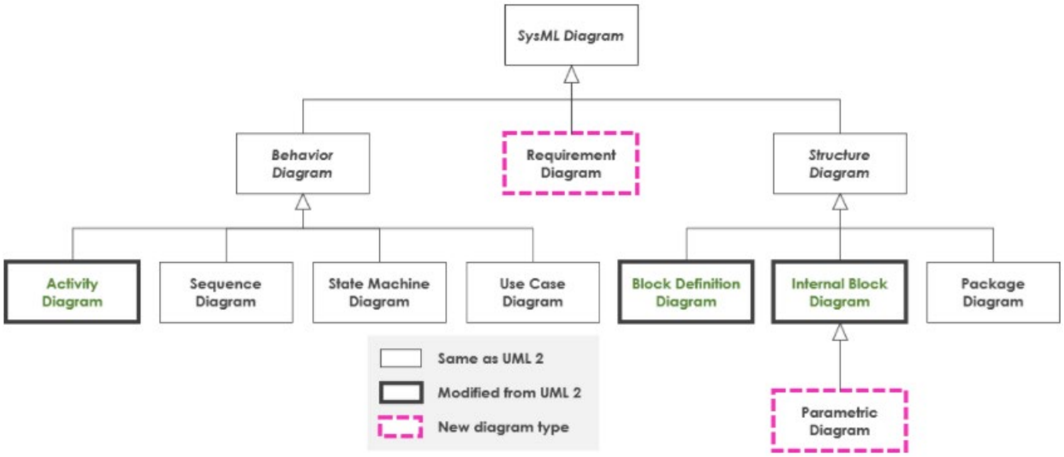


Figure 3. SysML Diagrams. Source: Visual Paradigm (n.d.).

Figure 3 shows that the activity diagram, block definition diagram (BDD), and internal block diagram(IBD) are SysML diagrams, but are modifications of UML2 diagrams, while the parametric diagram and requirement diagram are completely new creations specifically made for SysML. In general, the modification and addition of these diagrams can be seen as an overall improvement. Specifically, when comparing SysML to UML, the improvements include the following:

- SysML’s semantics are more flexible and expressive. SysML reduces UML’s software-centric restrictions and adds two new diagram types, requirements, and parametric diagrams.
- The requirement diagram can be used for requirements engineering;
- The parametric diagram can be used for performance analysis and quantitative analysis. (Visual Paradigm n.d.)

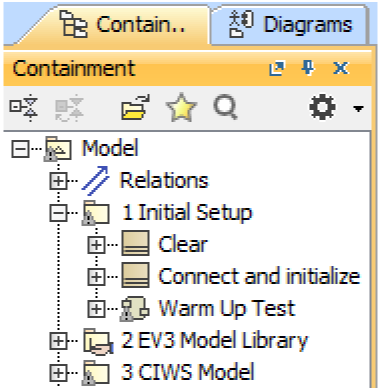
Given that SysML was created specifically for systems engineering purposes, these changes are sensible and align with the objectives of MBSE. Overall, SysML provides systems engineers with a language that can effectively model and describe the requirements, functions, and actions of a given system.

D. CAMEO ENTERPRISE ARCHITECTURE

No Magic created the first version of Cameo Enterprise Architecture in 1998 and is currently on the nineteenth version. Cameo's ability to incorporate a variety of different modeling languages while operating at a high level of sophistication has given it an advantage over competing software. According to the Dassault Systemes website, "Cameo Enterprise Architecture product, based on our core product MagicDraw, offers the most robust standards-compliant DoDAF 2.0, MODAF, NAF 3, NAF 4, and UAF 1.1 via a UAF standardized solution" (Dassault Systemes n.d.). Being able to support such a large amount of modeling operate languages allows for an increased ability to attract engineers searching for modeling tools. Other Cameo key benefits listed on the company's website include support of technologies, usability/user interface, adjustments/tailoring, distributed use/ parallel development, configuration management, requirements management, interoperability, and more.

Since the product was created there have been two main versions of Cameo Enterprise: a standard edition and an enterprise edition. Given that this research is mainly concerned with Cameo's SysML capabilities, the installed edition is the enterprise edition, which "contains all of MagicDraw's powerful UML, SysML, UPDM, BPMN, SoaML diagramming capabilities" (Dassault Systemes n.d.). Given the advantages of this edition, the standard edition was phased out and is no longer supported by any Cameo versions after 19.0. Regardless, the enterprise edition is sufficient and preferred for this research.

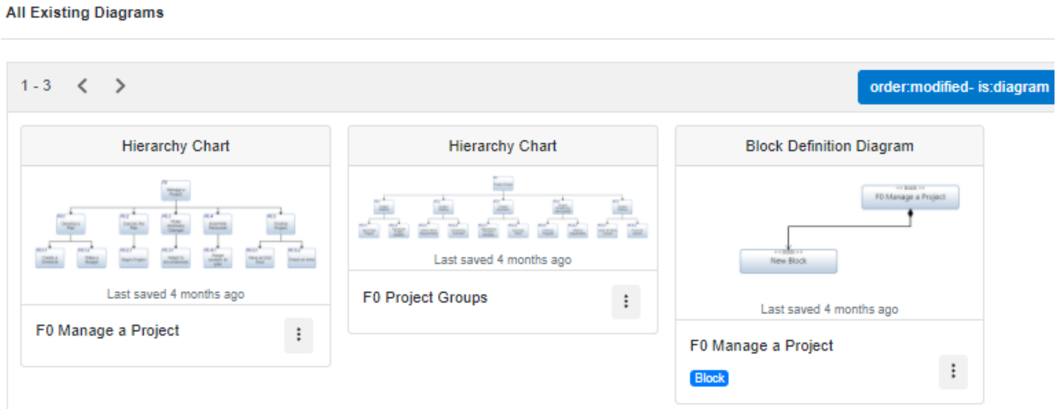
Cameo is not the only modeling tool in existence. Several other programs support many languages and perform similar functions to Cameo. Innoslate, for example, is a modeling tool that can support several languages, although not quite as many as Cameo, and can perform many necessary MBSE functions. However, there are key differences when it comes to other aspects of each modeling tool. One of the more deliberate differences with Cameo, when compared to Innoslate, is its organizational structure. Cameo utilizes a containment tree that can be organized into elements, such as a package, or different diagrams as seen in Figure 4.



- Model
  - Relations
  - 1 Initial Setup
    - Clear
    - Connect and initialize
    - Warm Up Test
  - 2 EV3 Model Library
  - 3 CIWS Model

Figure 4. Containment Tree from Cameo

This form of organization provides a view of the overall organization of a given project and allows for easy navigation between diagrams. On the other hand, Innoslate uses a database format and a wide view of all current diagrams, which might be easier to visualize as seen in Figure 5.



Hierarchy Chart
Last saved 4 months ago
F0 Manage a Project

Hierarchy Chart
Last saved 4 months ago
F0 Project Groups

Block Definition Diagram
Last saved 4 months ago
F0 Manage a Project
Block

Figure 5. Diagram Layout in Innoslate

However, switching between diagrams in Innoslate requires closing the current diagram and then reopening the desired diagram, whereas Cameo allows for multiple diagrams to be open at once and organizes them into tabs. Neither of these organizational differences is necessarily wrong or right. Rather, they demonstrate the different usability traits of each software and allow the user to choose either tool based on preference.

Aside from the organizational differences, Cameo allows users more access to the inner workings of the relationships and behaviors of different blocks, which in turn provides more detail as to how a system model, as a whole, is connected and how it operates. In general, Cameo provides more in-depth traceability throughout a given model, which makes it easier to connect system requirements to functions, as well as making it easier to navigate overall.

III. MK 15 PHALANX CIWS AND ITS MODEL

This chapter introduces the CIWS and how it is modeled in this thesis. First, there is background on the CIWS and its purpose in the fleet. Next, the Lego Mindstorms model is presented and its components are identified. Then, the process of integrating the Lego Mindstorms model with Cameo is discussed. Finally, the different physical connections within the model are described, as well as the inputs/outputs to/from both the Lego Model and Cameo model.

A. MK 15 PHALANX CIWS

As mentioned in Section I.B, the overall purpose of this research is to use the tools and principles discussed in Chapter II to create a dynamic model of a naval weapon system, specifically, Mk 15 Phalanx CIWS.

Figure 1 showed the two separate radars within the CIWS, as well as the gun barrels, and ammunition drums. The CIWS is deployed mainly on U.S. Navy cruisers and destroyers but can be installed on other platforms as well. After Soviet missiles sank the Israeli destroyer INS EILAT K40 in October of 1967, the world's navies realized they needed to focus on point defense systems At the time, defending against a "small, fast, low-flying threat represented by unmanned cruise missiles was something the U.S. Navy was not prepared for" (Stoner 2009). There were potential solutions that existed throughout the military, but a decision needed to be made on which type of weapon would be most effective. This led to General Dynamics, who "started with the Army's self-propelled M163 Vulcan Air Defense System to produce an automated version for Naval Service" (Stoner 2009). It appeared the best solution was simply to take a weapon that was used for defense against similar threats on the ground, and modify it so that it could operate on a naval ship. From this, the Phalanx was born and ships all across the navy would soon have one as a part of their defenses, with the first units being "installed aboard the USS Coral sea in 1980" (Stoner 2009). The weapon system has undergone several changes and upgrades over the years but still performs the same functions to this day.

The specifications for the CIWS can be seen in Table 1.

Table 1. CIWS Specifications. Source: Stoner (2009)

Gun	20mm M61! Vulcan Gatling rotary cannon
Height	15.4 feet (4.7 meters)
Weight	13,699 lb. (6,120 kg)
Elevation	=85 to -25 degrees at 115 degrees per second
Train	-150 to +150 degrees at 115 degrees per second
Working Circle	18 feet (5.5 meters)
Muzzle Velocity	3,650 fps (1,113 mps)
Rate of Fire	3,000 to 4,000 rounds per minute
Maximum Burst	1,000 rounds
Ammunition	1,550 rounds
Radar	Ku band search and fire control (tracking)
Cost	$5.6 million

From Table 1, there are a few specifications in particular that this research aims to model. The elevation, which represents the range of motion that the turret can move vertically, and the train, which represents the range of motion the base of the turret can rotate horizontally. These features can be modeled as constraints within Cameo so that the system can never exceed those certain boundaries.

The operational capabilities of the CIWS are search, detection, threat evaluation, tracking multiple targets, and engagement and kill assessment. CIWS is considered to be a last line of defense against incoming threats, which typically include "anti-ship missiles and aircraft, small high-speed surface craft, helicopters, and surface mines" (General Dynamics n.d.). These threats are generally summarized to either small, high-speed maneuvering craft or low, slow-moving craft.

CIWS can be controlled manually, but is designed to operate autonomously and can operate using its own radars and computing power, but can also be connected to Aegis and the Ship Self Defense System (SSDS). For this research model, it will be assumed that CIWS is operating using its built-in radars, given the modeling challenges of differentiating between self-operation and integration with larger defense systems. While in operating mode, the CIWS search radar is constantly looking for threats at great distances. If a threat is detected, it will track the threat until it crosses a certain distance threshold, at which point the tracking radar would take over. Once the threat comes within the final distance threshold, it is considered imminent and will be neutralized (Stoner 2009). This process can be modeled in Cameo using various diagrams and estimated distance thresholds.

B. LEGO MINDSTORM MODEL OF MK 15

The Lego Mindstorms model of the CIWS can be seen in Figure 6.

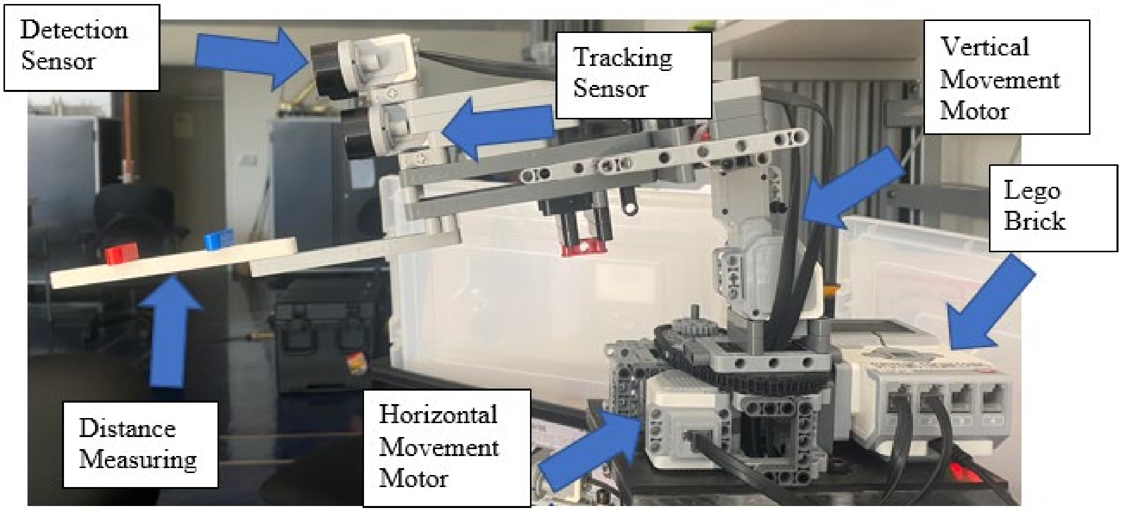


Figure 6. Full Lego Mindstorms CIWS model.

The model utilizes one motor to turn the turret horizontally and another motor to move the turret vertically, for a total of two motors. There are two different sensors attached to the barrel of the weapon. The top sensor is the detection sensor, and it is placed approximately one centimeter more forward of the tracking sensor to represent its longer range of detection. The bottom sensor is the tracking sensor which is meant for tracking the target as it gets closer to the ship. A flat Lego piece has been attached to the barrel to measure the target distances, and red and blue pieces represent the detection and tracking thresholds, respectively. The Lego Mindstorms “brick” is the controller for the model and can be seen in Figure 7.

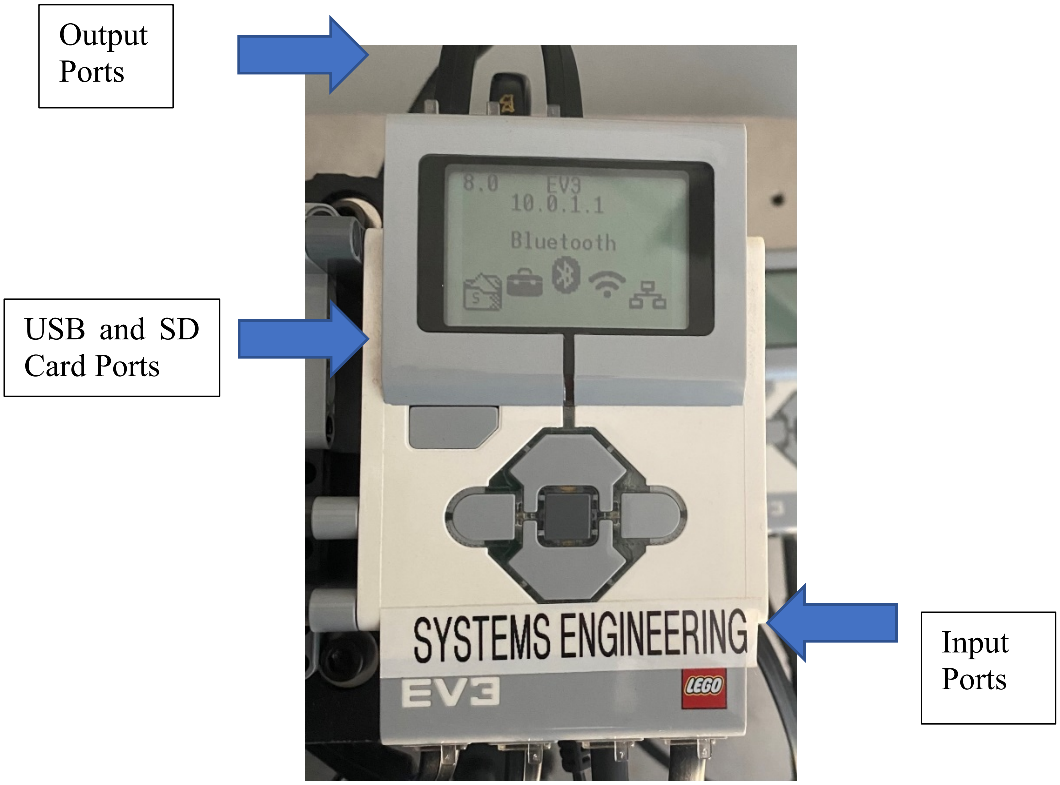


Figure 7. Lego Mindstorms Control “Brick.”

The brick has a built-in display which, due to the focus on Cameo, is not needed and is only used to establish the initial connection between the Lego model and Cameo. The brick can be powered using batteries but works most efficiently when using the rechargeable battery pack, which is constantly connected to wall power. Software can be implemented onto the brick from either a Universal Serial Bus (USB) connection, or with a Secure Digital (SD) card. As seen in Figure 7, the brick has four input connections at its bottom and four output connections at the top. These connections can be seen in Figures 8–10.

Figure 8. Lego Brick Input Ports.

Figure 9. Lego Brick Output Ports.

Figure 10. Lego Brick USB and SD Compatibility.

Lego has its software built into every Lego brick, but this software cannot be integrated with Cameo. To combat this, software called Lejos EV3 is downloaded onto an SD card and inserted into the brick, which will then run instead of the default Lego software. Lejos is a Java Virtual Machine that was configured to run on the Lego EV3 brick that can communicate directly with the Lego EV3 Plugin for Cameo and allows Cameo to receive inputs from the Lego sensors and give outputs to the motors. Detailed instructions for downloading the Cameo EV3 Plugin and the Lejos EV3 operating system replacement were provided by a blog created by thesis co-advisor Dr. Saulius Pavalkis (2019). The specific connections between the Lego model and Cameo are discussed in the next section.

C. INPUTS AND OUTPUTS

Given the various components of the entirety of this model, it is important to have an understanding of the various connections within the system. Figure 11 shows an internal block diagram, created using Cameo that identifies the major system parts and the connections between them.

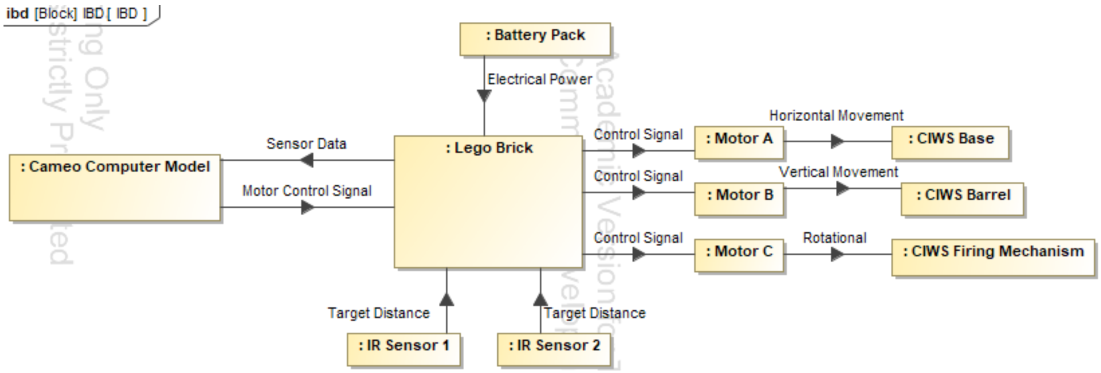


Figure 11. Internal Block Diagram of CIWS Model.

The block on the left side of Figure 11 represents the computer model, which can receive sensor data from the Lego brick and send control signals to the brick, all via a Bluetooth Personal Area Network (PAN). The Lego brick receives the target information from the IR sensors, and sends it to the Cameo model, while also sending the Cameo control signals to the motors. These signals move the motor to the desired position, based on the sensor input. Motor C is simply a small motor added to represent the firing of the weapon and simply rotates for a set amount of time when prompted. The battery pack provides power to the Lego brick, which then distributes this power to the motors and the sensors, allowing the entire model to operate without being close to a wall socket. Understanding the different inputs and outputs of a system is a critical skill to have as a systems engineer because when it comes to integrating complex software and hardware, the connections can become unorganized. Having a firm grasp on the different connections within a system and between systems is beneficial when performing integration and overall helps to avoid potentially problematic errors.

D. MODELED SCENARIO

The main threats the CIWS is meant to defend against include either small, high-speed maneuvering craft or low, slow-moving craft. These typically include anti-ship cruise missiles, aircraft, or small surface vessels. Assuming the CIWS is in the idle position, which indicates it is vertically and horizontally level, this will be the reference point. The target is inbound directly at the CIWS and passes the detection threshold. The CIWS moves horizontally to Angle A and vertically to Angle C, which is where the target has moved. It is important to note that typically, CIWS does not move its barrel until in tracking mode, but given the placement of the sensors in the model and their limitations, the assumption is that the CIWS adjusts to point the detection sensor towards the target while it is inbound. Once the target crosses the engagement threshold, the turret will move horizontally to angle B and vertically to Angle D. The starting and final positions of the target are shown in Figures 12 and 13.

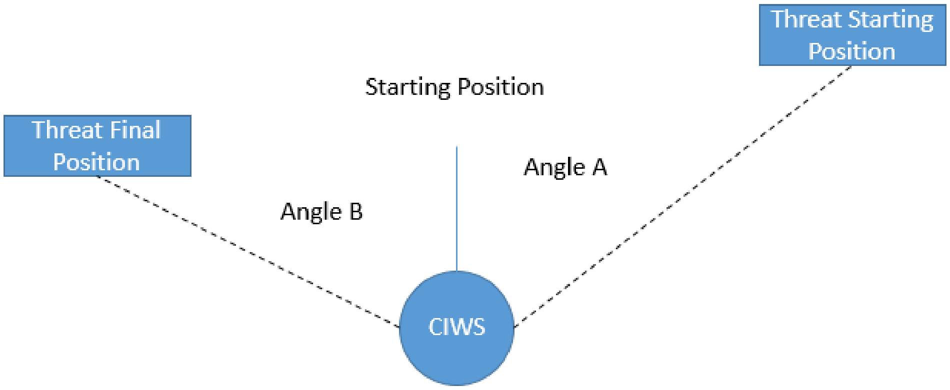


Figure 12. Top-Down View of CIWS and Inbound Threat with Horizontal Movement Angles.

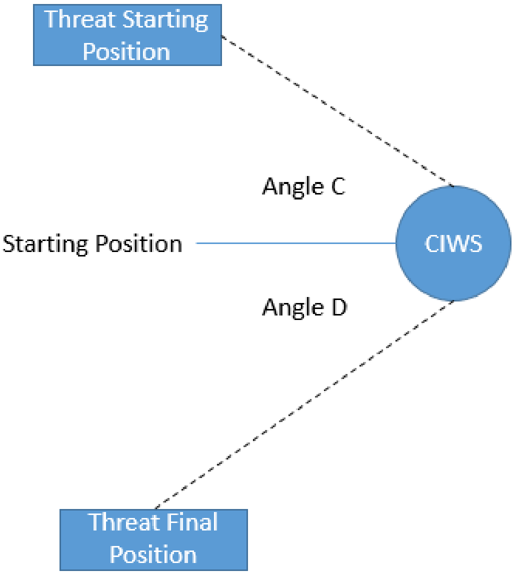


Figure 13. Side View of CIWS and Inbound Threat with Vertical Movement Angles.

Due to software limitations, the angles for the model are entered manually and are therefore arbitrary. The angles chosen need only not exceed the CIWS horizontal and vertical movement limitations displayed in Table 1. Table 2 gives the specific angles that were used for this modeling scenario.

Table 2. Angle Values Used for the Model

Variable	Value (degrees)
Angle A	+30
Angle B	-50
Angle C	+20
Angle D	-10

The angles in Table 2 are all relative to the initial starting point when the turret is in the idle position. However, when the Cameo model receives angle inputs, it moves the motor by the specified angle value, as opposed to moving the motor to an angle. Therefore, the angles that enter into the model must be the total degrees needed to move the motor position from one angle to another. For example, to move from Angle C to Angle D, the motor must receive an input of +30, which would move the motor downwards 30 degrees. Due to the configuration of the Lego Motors, a positive input moves the motor downwards and vice versa. This will be discussed further in Block Definition Diagram in Chapter IV.

CIWS operates using different thresholds for its search and tracking radars. The search radar has a longer range, which is modeled by the top infrared sensor being slightly more forward than the bottom infrared sensor, and "the search radar identifies the target at 10 nautical miles and the software begins tracking" (Stoner 2009). For the tracking radar, "target assignment and priority are done at 5 miles and engagement begins at or about 2 miles" (Stoner 2009). These values are to be compared to sensor values and then used to insert these thresholds within the model. Figure 6 displayed red and blue Lego pieces that were placed in front of the sensors to represent the detection and tracking thresholds. When the target is beyond the red piece, it is considered to be within the detection, or search, threshold. Once it is within the red piece, it is close enough to activate tracking mode. Finally, once the target comes within the blue piece, it is within proximity to the ship and will be destroyed. The values for these thresholds are shown in Table 3.

Table 3. CIWS States and Corresponding Thresholds

CIWS State	CIWS Threshold (mi)	IR Sensor Threshold (cm)
Detection	< 10	< Max reading
Track	< 5	< 14
Engagement	< 2	< 10

According to the infrared sensor documentation, while in proximity mode, the sensors should return values from 0 to 100. Lego does not provide a unit for the values and they do not represent specific distances, but rather "will depend on the strength of the signal and other factors" (Lego Mindstorms EV3 Help n.d.). However, while testing the sensors, this range was not quite the same. The maximum value the sensors would return was between fifty and fifty-five, and anything beyond that returned the value of negative one, which means the sensors were not detecting the target. This complication, along with the overall sensitivity of the sensors, made it difficult to accurately represent the realistic distances. To compensate for this, data values were collected and analyzed to create a conversion between the sensor reading and a distance in centimeters. Sensor readings were collected at every centimeter between four and twenty-one centimeters. The results were used to calculate the conversion shown in equation 1.

$$y = 0.003x^2 + 0.3901x + 3.0951 \tag{1}$$

The x value in equation 1 represents an inputted sensor value, and the y value is the outputted distance in centimeters. From here, for modeling purposes, the threshold values were determined by taking multiple sensor readings at each threshold point and averaging the results. These average values represent what value each sensor should be reading when the target is at the threshold limit, and are the most effective value to use for the model thresholds. These values were then converted to centimeters, and the thresholds were determined using these results. Although they do not directly correspond with the real weapon thresholds, they are sufficient thresholds that can be used effectively in the model.

Given this scenario, along with the angle values and thresholds, the next step is to model the system in Cameo.

IV. MK 15 MODELING IN CAMEO

Chapter III described the physical model of the CIWS that was constructed using the Lego Mindstorms, as well as the scenario that is being modeled. This chapter introduces the diagrams that were generated to create a digital twin of the CIWS in Cameo. Several diagrams from SysML were used including block definition diagrams, activity diagrams, and state machine diagrams. The advantage of using Cameo is that once the model is complete, the Cameo Simulation Toolkit can be used to execute certain aspects of the model and simulate realistic scenarios.

A. BLOCK DEFINITION DIAGRAM

A Block Definition Diagram (BDD) is the most effective diagram to use when organizing a system's overall structure. Typically, this refers to physical aspects of the structure, but can also be used to show how the system interacts with other components. An engineer should be able to look at a BDD and have a general understanding of all the different physical components that make up a system as well as their overarching relationships with the system as a whole. Figure 14 is the BDD for the CIWS that was created using Cameo.

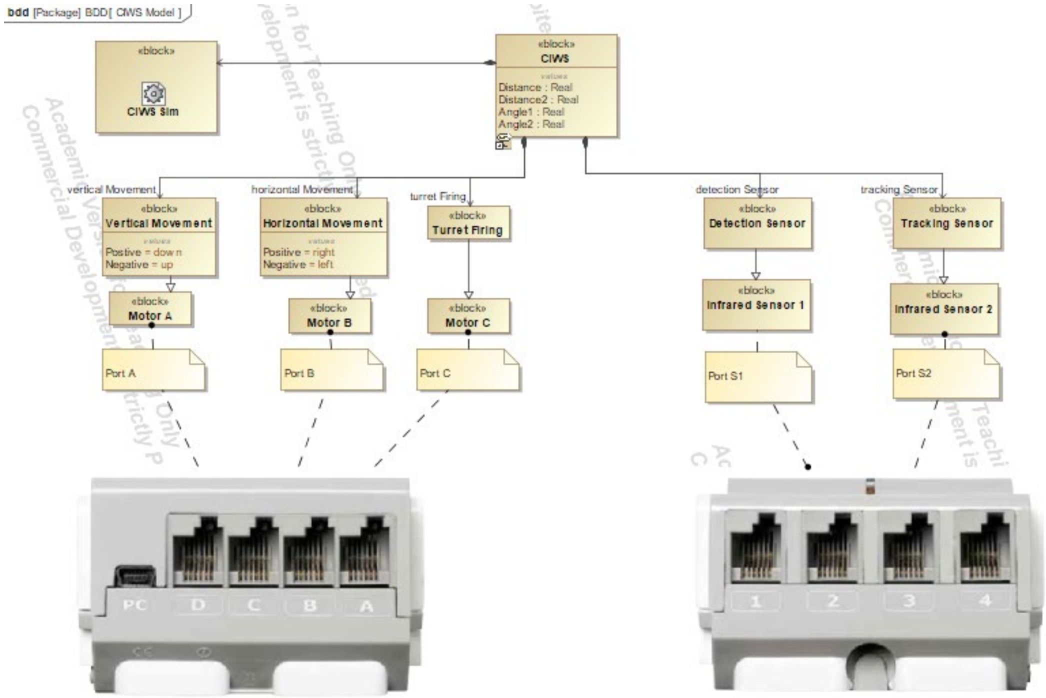


Figure 14. CIWS BDD Created in Cameo.

The top-level block shows the CIWS and the values that are used throughout the model. The left side of Figure 14 has blocks for vertical and horizontal movement, which are then broken down into the specific motor and port that is used to control each respective movement. Figure 15 shows an enlarged version of these blocks.

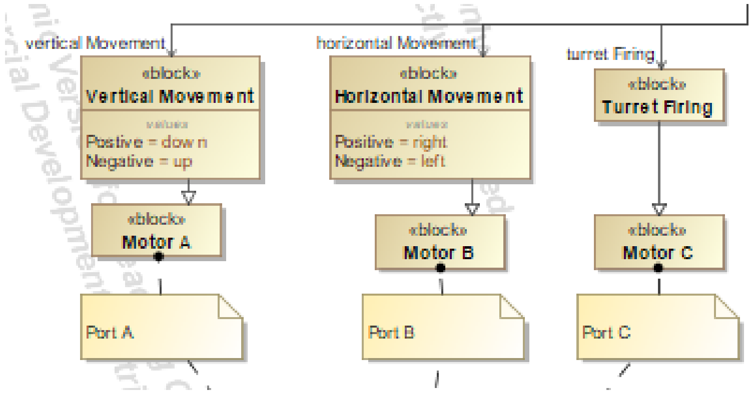


Figure 15. CIWS Model Movement as Shown in BDD.

The values written inside the movement blocks reference the direction the motors will move given the corresponding sign of the input angle. For example, if a positive angle is sent to port A, the model moves downwards, and if a negative angle is sent to port B, the base of the CIWS moves to the left, or counterclockwise looking top down. Motor C is a small motor that spins rapidly upon command, to represent the spinning of the gun barrel. The right side of the BDD in Figure 14 is shown in Figure 16.

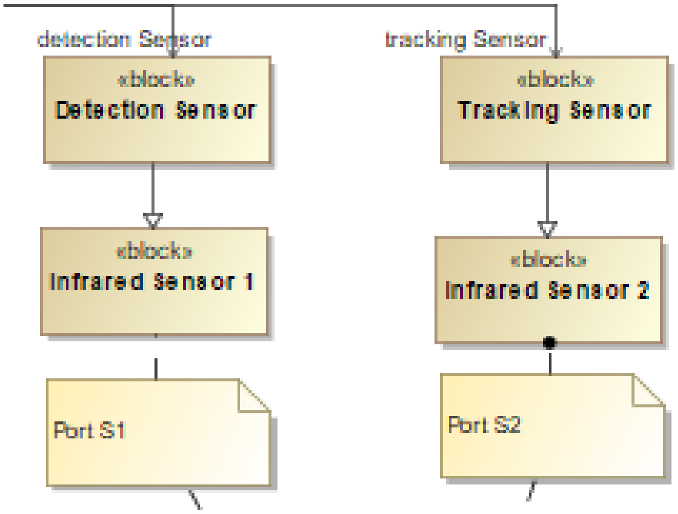
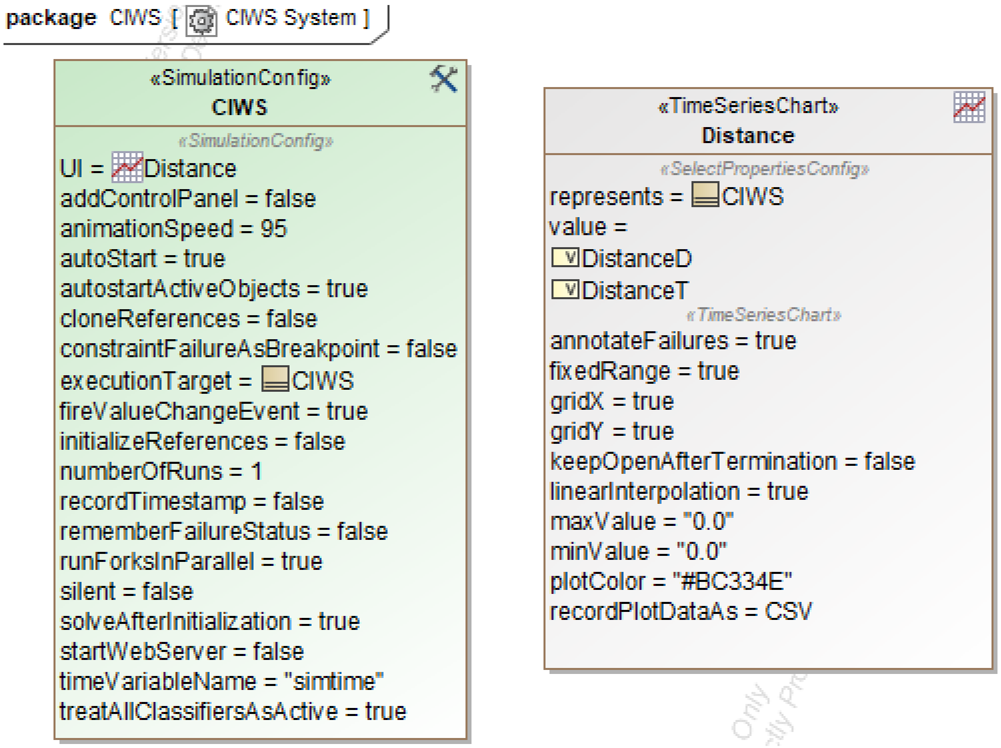


Figure 16. CIWS Sensor Models as Shown in BDD.

The Lego Mindstorms model of the CIWS uses two infrared sensors to represent the detection and tracking sensors. Figure 16 shows that infrared sensor 1, which is connected to input port S1, directly represents the detection sensor, and infrared sensor 2 represents the tracking sensor. The top left corner of Figure 14 includes a block labeled CIWS Sim, which is where the model simulation is configured. The details of how this simulation is displayed are shown in Figure 17.



«SimulationConfig»
CIWS
«SimulationConfig»
UI = Distance
addControlPanel = false
animationSpeed = 95
autoStart = true
autostartActiveObjects = true
cloneReferences = false
constraintFailureAsBreakpoint = false
executionTarget = CIWS
fireValueChangeEvent = true
initializeReferences = false
numberOfRuns = 1
recordTimestamp = false
rememberFailureStatus = false
runForksInParallel = true
silent = false
solveAfterInitialization = true
startWebServer = false
timeVariableName = "simtime"
treatAllClassifiersAsActive = true

«TimeSeriesChart»
Distance
«SelectPropertiesConfig»
represents = CIWS
value =
DistanceD
DistanceT
«TimeSeriesChart»
annotateFailures = true
fixedRange = true
gridX = true
gridY = true
keepOpenAfterTermination = false
linearInterpolation = true
maxValue = "0.0"
minValue = "0.0"
plotColor = "#BC334E"
recordPlotDataAs = CSV

Figure 17. CIWS Simulation and Time Series Block.

The block on the left is used to configure what aspects of the model are being simulated, and how the inputs and outputs are going to be displayed. In this case, the outputs are modeled by a time series chart, as shown by the block on the right of Figure 17. Together, these blocks create both a user interface and a way for the data from the model to be displayed.

The BDD is used mostly for organizational purposes and to give a general idea of how different components are related to the system as a whole, not necessarily with each other. The IBD in Figure 11 is used as a more in-depth description of the relationship between system components, including how they are connected and what type of data is passed between them. Using the BDD as a reference, the next step of creating the model is to create an executable state machine diagram to actuate the model.

B. STATE MACHINE AND ACTIVITY DIAGRAMS

Creating a dynamic model means that the system is moving between different states based on certain inputs from the surrounding environment. Therefore, a state machine diagram is used to effectively model these states and the transitions between states. Within the state machine diagram, activity diagrams will be called upon to receive inputs from the infrared sensors, and then send corresponding signals to the motors. Together, these diagrams create an automated model of the CIWS. The state machine diagram in Figure 18 represents the CIWS and the different states that are modeled.

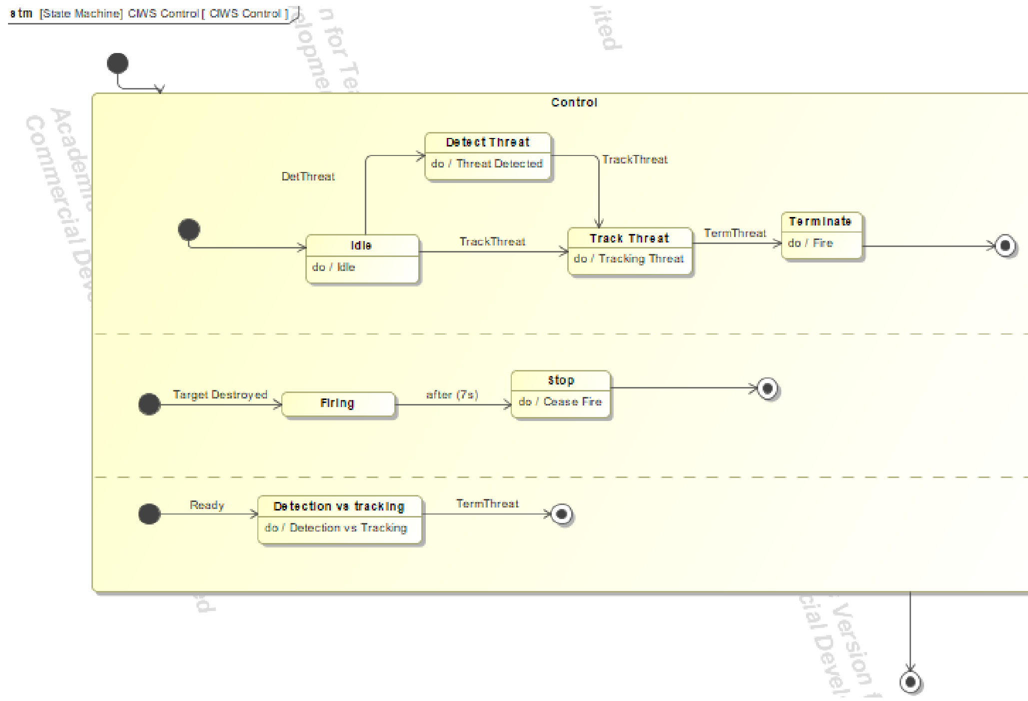


Figure 18. State Machine Diagram Used to Model CIWS.

Each block in Figure 18 represents a different state the CIWS could be in at any given moment. The solid black nodes are the starting points used for when the simulation is executed, and the partially filled nodes are the final nodes that end the simulation. For example, the black node at the upper-left executes the control block as a whole, and then each of the three sections begins being executed simultaneously by their respective starting nodes. The connections between each block are known as “transitions,” given that they are how the model transitions between states. Transitions can be labeled in many ways, but the labels in this diagram represent signals. A signal can be sent at specific points in the model, which then activates corresponding transitions. These transitions execute the corresponding state, which continues to execute until another transition is signaled. Signals and transitions are a great way to automate the model using input from sensors to create thresholds and corresponding command outputs to the motors. Figure 19 breaks down the first section of the state machine diagram.

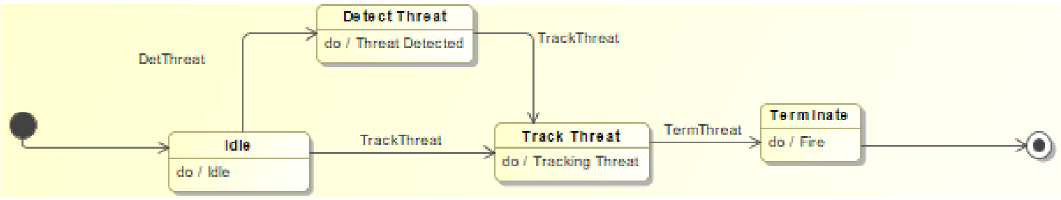


Figure 19. Output Section of the State Machine Diagram.

The section shown in Figure 19 represents the different states that are outputted to the CIWS motors. The initial idle state moves the CIWS to a position where it can begin searching for threats. From the idle state, Figure 19 shows two signals that can lead to a transition. The DetThreat signal transitions the model to the Detect Threat state and the TrackThreat signal transitions the model to the Track Threat state. Within the state blocks, certain actions are called upon when the state is executed. These actions represent activity diagrams that actuate the Lego model based on received inputs. Figure 20 shows the activity diagram for the idle state.

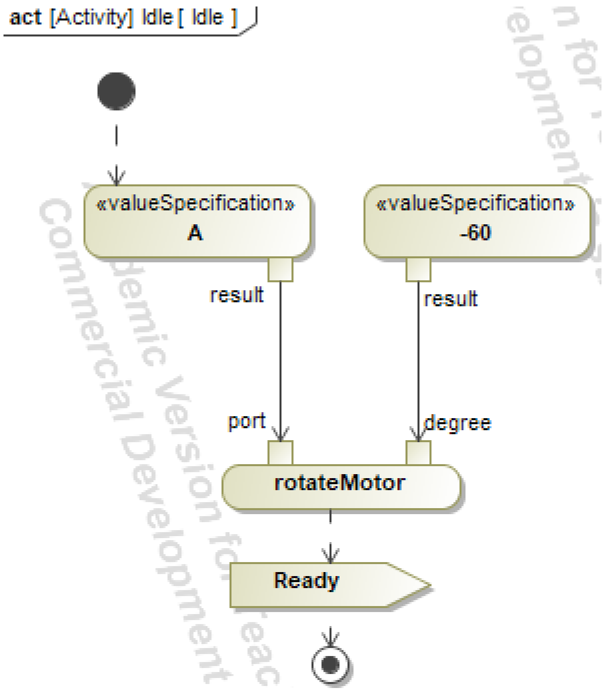


Figure 20. Idle State Activity Diagram.

When the Idle state is executed, the activity diagram in Figure 20 uses a valueSpecification block to declare the port value and angle to move the motor. These values are inputs to the rotateMotor block, which comes directly from the EV3 Model library in Cameo, and then moves the motor A to the specified angle, which in this case is negative sixty. It was determined through testing that negative sixty was the best idle position, given that it moved the turret to a level vertical position. The block labeled "Ready" is the signal block, which when activated at the end of the activity, sends the ready signal to activate a transition within the state machine. In this case, the ready signal is sent to the third section of the state machine diagram, shown in Figure 21.

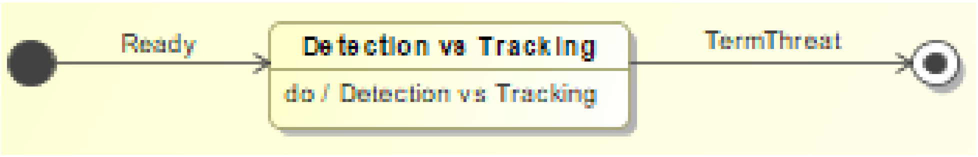


Figure 21. Sensor Input Section of the State Machine Diagram.

The ready signal from Figure 20 is shown as the transition to the Detection vs Tracking block in Figure 21. This block is where inputs are received from the infrared sensors and then used to determine whether the CIWS model should be in the detection or tracking state, which is based on the values described in Chapter III. Figure 22 displays the activity diagram for the Detection and Tracking state.

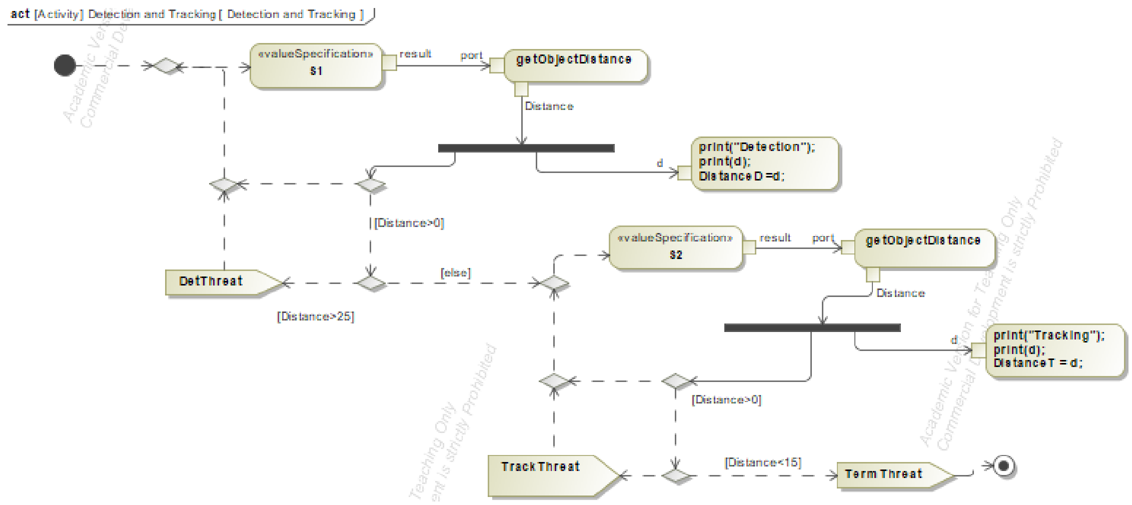


Figure 22. Detection vs Tracking Activity Diagram.

Given that the Detection vs Tracking state is important for determining other states in the model, it is highly complex. This model was created to best represent how the radars on the CIWS operate. The detection radar has a longer range and is used for the initial detection of threats. The tracking radar has a shorter range and is more effective at tracking the targets as they move within proximity to the ship. As a target approaches the ship, the detection radar continues to monitor the threat but passes on the target to the tracking radar as it crosses predetermined thresholds.

There are essentially two main loops within the activity diagram in Figure 19. The first loop is the detection loop, as shown in Figure 23.

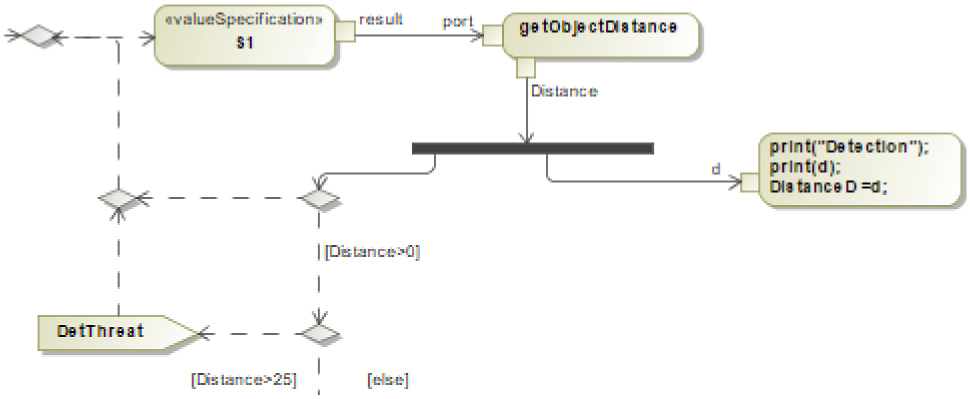


Figure 23. Detection Loop.

It is important to note that the dotted lines in the activity diagram are known as control flows, which do not pass data, but only determine the order, or flow, of which blocks are to be executed. The solid lines are called item flows and pass items, or other forms of data, from one block to another. This loop shows port S1 is specified, which corresponds to the infrared sensor 1 and then used as an input to the getObjectDistance block, which also comes from the EV3 Model library in Cameo. This block outputs the current distance to the target, as read by infrared sensor 1. This distance is passed to the opaque action block, which allows for code that prints values to the screen and assigns the variable DistanceD to be executed. The distance is also passed to an initial decision block, which is primarily used to filter out any values below zero, which appear when the sensor is not detecting the target. The next decision block determines if the threat is outside the first threshold, which is set as 25, or 14 centimeters. If the distance is greater than this threshold, the DetThreat signal is activated, and the loop is restarted. This loop continues to run until the target crosses the threshold, which then activates the tracking loop shown in Figure 21.

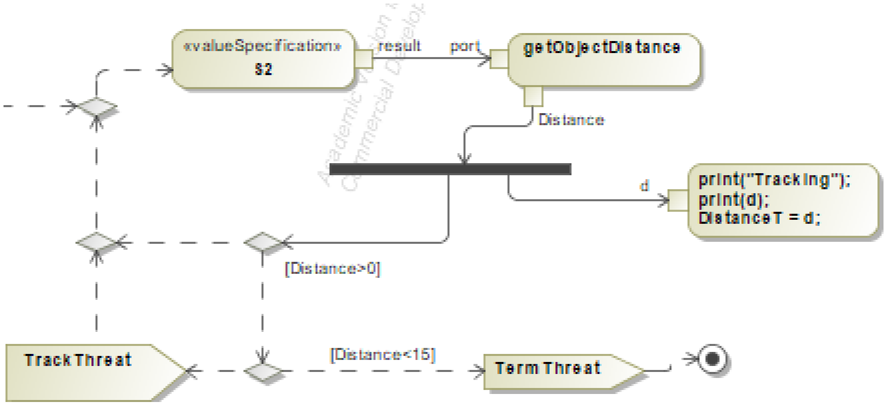


Figure 24. Tracking Loop.

As shown by the valueSpecification block, the inputs for this loop are now coming from the S2 port, which corresponds to infrared sensors 2. Similar to the detection loop, the distance is read and printed, as well as saved as the variable DistanceT. The inadequate values are filtered out by the first decision node, and the second decision node determines if the target has crossed the next threshold, which is 15 in this case. While the target is between the distance 25 and 15, which corresponds to 14 and 10 centimeters, the loop continues to run and the TrackThreat signal is sent and the Tracking Threat activity diagram is executed. When the distance becomes less than 15, this means the target is near the ship and needs to be terminated. This activates the TermThreat signal which ends this activity diagram and executes the firing block.

The activity diagram in Figure 19 outputs two different signals to the state machine diagram based on inputs from the sensors. Figure 16 shows that the DetThreat signal transitions the state machine diagram from Idle to the Detect Threat state, which then executes the activity diagram in Figure 22.

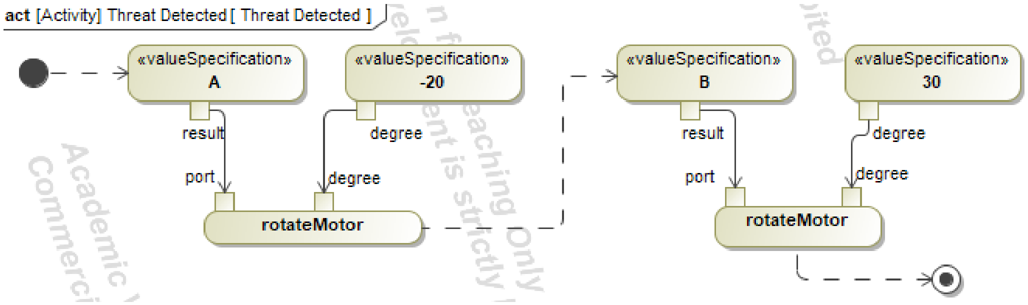


Figure 25. Threat Detected Activity Diagram.

The execution of this diagram results in the movement of both motors A and B, as shown by the blocks in Figure 25. They are moved to the detecting position, which is determined by the scenario given in Section 3.D. As shown, this diagram is not a loop and therefore only moves the motors once when called upon to do so. Figure 18 showed that the motors remain in this position until the TrackThreat signal is sent, which occurs at the end of the tracking loop in Figure 24. This represents the target crossing the final distance threshold, meaning it is close enough to the ship that it needs to be destroyed, and transitions the state machine to the Track Threat state. Figure 26 displays the activity diagram corresponding to this state.

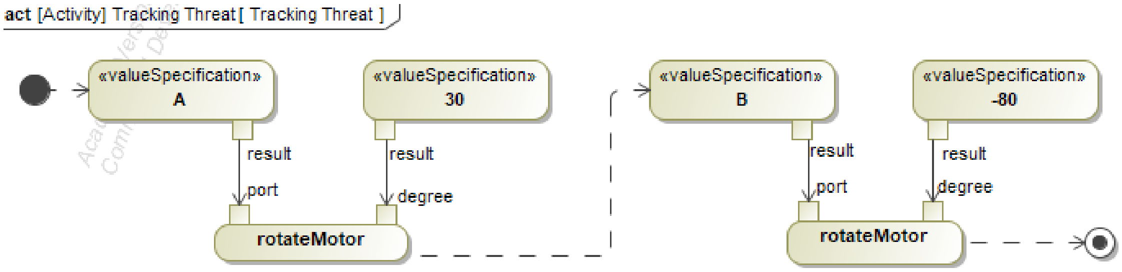


Figure 26. Tracking Threat Activity Diagram.

Similar to Figure 25, this diagram moves both motors A and B to the tracking and final position, which corresponds to the position from the scenario described in Section 3.D. Once the target crosses the engagement threshold and the TermThreat signal is sent, this activity diagram ends, and the Terminate Threat activity diagram, shown in Figure 27, is executed.

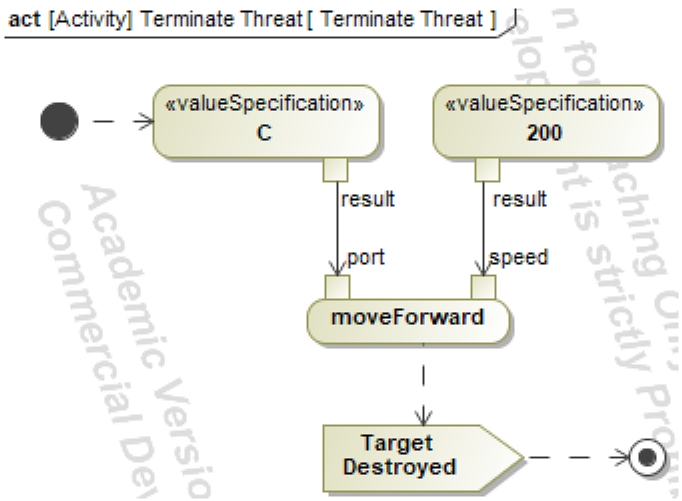


Figure 27. Terminate Threat Activity Diagram.

This activity diagram rotates a small motor connected to port C, which represents the physical firing of the weapon system. The Target Destroyed signal activates the final section of the state machine, which stops the rotation of the firing motor and completes the model execution as shown in Figure 28.

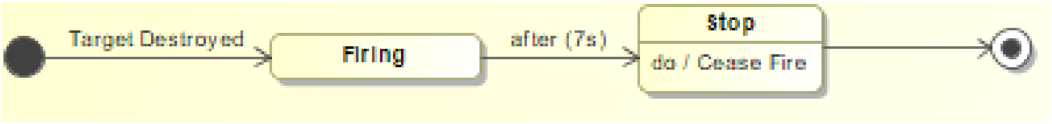


Figure 28. Terminate Threat and Final Section of State Machine Diagram.

The activity diagram for the Stop state is shown in Figure 29.

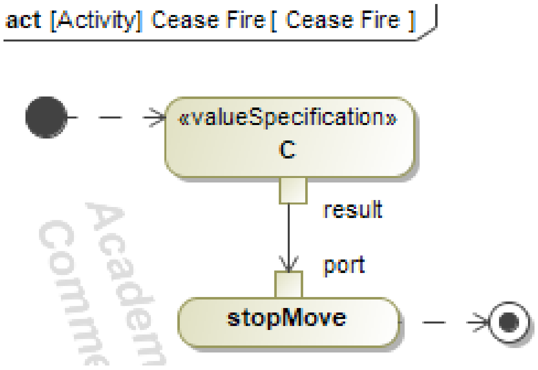


Figure 29. Cease Fire Activity Diagram.

This activity diagram stops the firing of the turret using the stopMove block, which represents the destruction of the target and the completion of the executable model.

C. SIMULATION OUTCOME

As the threat, modeled using the infrared beacon, moves around the modeling environment, the Cameo model reacts according to the data from the infrared sensors and moves the position of the turret accordingly. Figures 30–32 show the different positions of the turret as the model is executed.

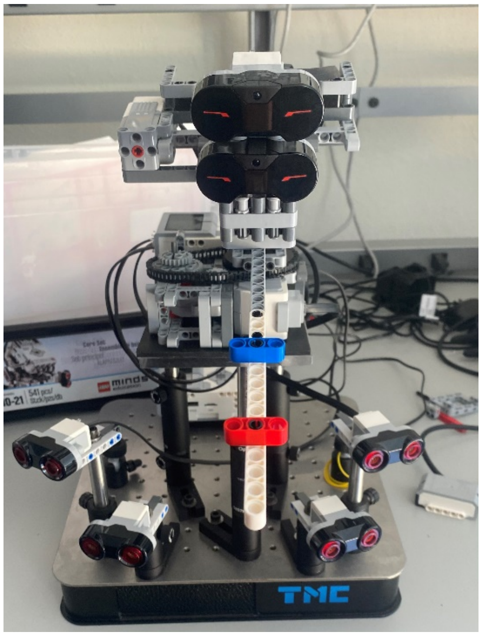


Figure 30. Idle Position.

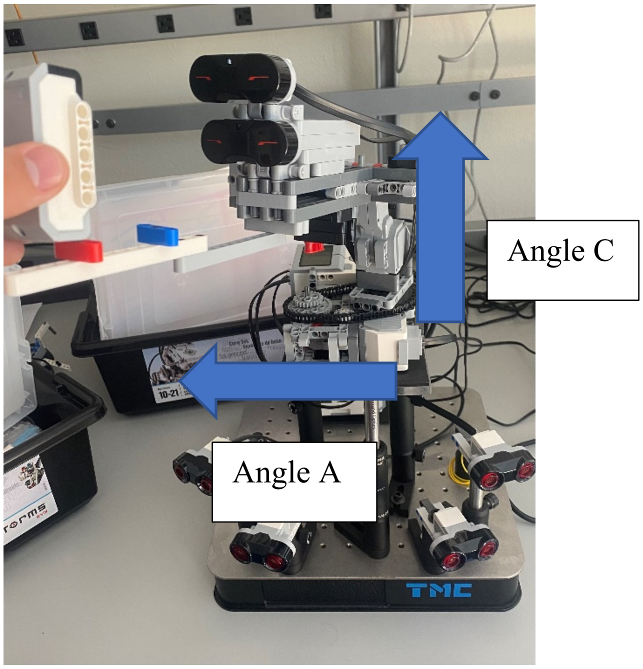


Figure 31. Detection Position.

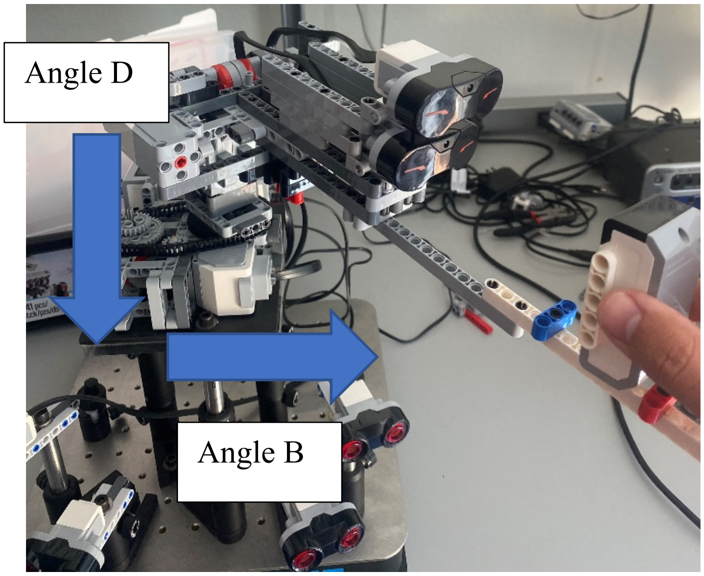


Figure 32. Final Threat Position.

Figures 30–32 show that the model is capable of taking information from the sensors, using it in the model, and then moving the motors to the corresponding positions.

Figure 33 shows how both sensors tracked the distance compared to the time in the simulation.

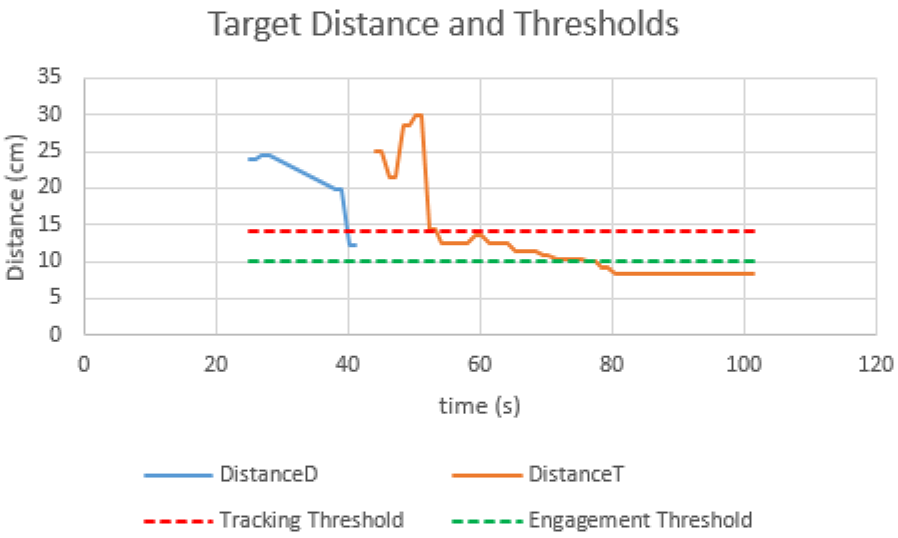


Figure 33. Detection and Tracking Distance.

The target is initially detected by the detection sensor just after 20 seconds at a distance of approximately 25 centimeters, as shown by the blue curve. The blue curve represents the data inputs from the detection sensor, which ends after going below the red dashed line at around 40 seconds. The red dashed line represents the tracking threshold, which when crossed, would switch the inputs from the detection sensor to the tracking sensor. The orange curve represents the input from the tracking sensor. There are a few initial high values while the sensor first picking up the target, but it levels out as the target moves from a distance of 14 centimeters to 10 centimeters. The green dashed line represents the engagement threshold at a distance of 10 centimeters, and once the target crosses this line just before 80 seconds, the model is given the command to fire and the target is considered destroyed. It is important to note that the time in this figure begins just after 20 seconds due to all values before that being invalid, which occurs when the sensors are first locating the target.

D. CURRENT LIMITATIONS OF THE INTEGRATED MODEL

As shown in Section 4.C, once initialized, the model can read the specified sensor data and move the motors to predetermined angles based on the input. However, the model does not reach its full potential due to limitations with the Cameo and Lego Mindstorms software integration. The first limitation comes from the infrared sensors. These sensors are designed to have several different modes such as proximity mode, beacon mode, and remote node (Lego Mindstorms EV3 Help n.d.). These modes allow for the sensor to change how it receives its input. The EV3 Cameo library does operate on both proximity and beacon mode, meaning it can pick up the signal from the beacon if necessary, and in beacon mode, the beacon proximity or heading can be given. However, for undetermined reasons, the EV3 library is not able to switch between the proximity and heading setting within beacon mode. It only inputs the distance to the target with no indication of direction. It is for this reason that the angles that the motors move to need to be predetermined based on the given scenario. The ability to switch between modes is done easily using the Lego Mindstorms software but has yet to be incorporated into Cameo.

The other model limitation involves being able to read the current angle of the motor. The EV3 library has several functions to change both the speeds and angles of the motors. However, there currently is not a function in Cameo to read the angle of the motor at any given time, which is available with the Lego Mindstorms software. The ability to read the motor angle combined with using infrared sensors to determine target heading would allow for real-time target tracking. It would require that code be developed to read all the necessary data, and a controller could be created to effectively actuate the model.

Given these limitations, the scenario outlined in Section 3.D was based on a predetermined target path and therefore predetermined target angles. Representing the scenario in this fashion removed some of the realism from the model, but still provided the opportunity to experiment with modeling a naval weapon system using the concept of digital twins and advanced software from Cameo.

V. CONCLUSION

The previous chapter displayed how the system was modeled in Cameo and the simulation outcome. This final chapter summarizes this research, discusses whether or not the research questions were answered, and explores future applications of this research.

A. SUMMARY

The goal of this research was to explore advanced systems engineering software to create a dynamic model of a naval weapon system. The motivation behind this research came from a recent push to incorporate digital engineering and the use of digital twins in the DOD's acquisition process. The weapon chosen for this research was the Mk 15 Phalanx CIWS, a turret-like weapon system deployed on numerous U.S. Navy ships. A physical model of the weapon was configured using a Lego Mindstorms EV3, and the Lego controlling brick was configured so that it could be integrated with Cameo Enterprise Architecture.

Exploring the Cameo software was the main focus of this research, given its advanced modeling and simulation capabilities. Several reasons as to why the software is superior to other competing software were discussed. A model was created in Cameo using several different SysML diagrams. These diagrams, when combined and utilized effectively, create a dynamic model that can receive inputs and create control signals based on these inputs. Several limitations affected the scope of the project. However, these limitations were due mostly to a lack of functions that were developed for the EV3 library in Cameo. The Cameo software was unable to access the full capabilities of the Lego hardware, and therefore decreased the overall capabilities of the model. Despite limitations, the model performed its most basic functions and provided a foundation for further testing in the future.

B. KEY FINDINGS

The first research question formulated in Section I.C was whether Cameo Enterprise Architecture could be used to model and control systems created using Lego

Mindstorms. The research showed that the Lego Mindstorms could be integrated effectively with Cameo. Once integrated, the system could be modeled using a variety of SysML diagrams alongside the EV3 library that was developed for Cameo. This library included several functions that were used to send and receive information to and from the Lego hardware. Given the model is configured correctly, it can be executed and the Lego Mindstorms sensors receive inputs that then affect the signals the model outputs. Due to the discussed limitations (see Section IV.D), the model is not considered to be fully autonomous. There is a small level of autonomy built within the model and all of the outputs are entered manually and predetermined, as opposed to constantly changing in response to the inputs. If Cameo were able to access all of the capabilities of the Lego Mindstorms hardware, then there would be a possibility of creating a system that is much more autonomous. Overall, it was determined that Cameo could be used to model and control systems using Lego Mindstorms, but only to a certain extent given current limitations.

The second research question was whether the integrated Cameo/Lego Mindstorm tool is useful to conduct parametric and trade-off studies and to verify system-level requirements. The objective behind creating a digital twin was to use the model to conduct different studies to retrieve data from the system in a cost-effective and less complicated manner. Instead of having to take the system out of operational capacity to conduct tests, the system can remain operational while the tests are instead conducted on the digital twin, therefore saving money. The constraints that were discussed throughout this research made it difficult to create a model that could then be used for parametric studies. The need to use predetermined output values and inputs received from manual changes to the environment meant that there was little to no room for changing system parameters to then study different outputs. Had the Cameo EV3 library provided more capabilities, the system would have contained less predetermined values and instead allowed for an increase in overall autonomy. This increase in autonomy would have enhanced the ability to perform parametric and trade-off studies and the ability to verify system requirements. Overall, the limitations of the Cameo EV3 library did not allow for parametric studies to be conducted, however future software updates should be able to mitigate this problem.

C. FUTURE WORK

An increase in the capabilities of the Cameo EV3 library is expected to provide the opportunity for more in-depth model configuration in future research. The models will be able to operate more autonomously due to an increase in the level of control the Cameo software will have over the Lego hardware. Further complex integration could lead to more realistic models that allow for extensive studies to be conducted in safe environments away from the battlefield. This research can be applied to numerous weapons systems throughout the navy and the DOD as a whole. By increasing the ability of engineers to effectively model systems using digital twins, the DOD would save time and money in several ways throughout the acquisition process.

The use of a Lego Mindstorms kit worked well for this research, given that overall, it was compatible with Cameo. However, this research does not have to be limited to Lego Mindstorms. There are more advanced ways to create models of systems that may vary in size and scale. Using similar methodologies that were utilized in this research, future engineers could produce complex models with the ability to accurately replicate realistic scenarios and simulate how the system should react.

The use of digital twins does not only apply to military weapon systems but can also be used on other systems throughout the DOD. Entire warships could be modeled and used for parametric testing, as well as specific parts of the ship such as navigational equipment or engine parts. Military ground vehicles could be modeled on a smaller scale and tested using the concept of digital twins, without having to take away a vehicle that could be made operational. This research could even be applied to advances in swarm technology. Multiple digital twins could be designed to replicate any type of swarm, and system analysis of these models could be performed. Overall, there are many different applications of this research that should be explored through further research and experimentation.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Buede, Dennis M. 2009. *The Engineering Design of Systems:Models and Methods* (2nd edition). Hoboken, NJ: John Wiley and Sons Inc.

D. Kaslow, G. Soremekun, H. Kim and S. Spangelo, 2014. "Integrated model-based systems engineering (MBSE) applied to the Simulation of a CubeSat mission," 2014 IEEE Aerospace Conference, Big Sky, MT, USA, pp. 1–14, doi: 10.1109/AERO.2014.6836317.

Dassault Systemes. n.d. "Cameo Enterprise Architecture" Accessed May 1, 2021. https://www.3ds.com/products-services/catia/products/no-magic/cameo-enterprise-architecture/

Finance, Guillaume. 2010. "SysML Modelling Language explained." October 7, 2010. https://www.omgsysml.org/SysML_Modelling_Language_explained-finance.pdf

General Dynamics. n.d. "Naval Platform Systems: Phalanx CIWS." Accessed May 1 2021. https://www.gd-ots.com/armaments/naval-platforms-system/phalanx/

International Counsel on Systems Engineering. n.d. "What is Systems Engineering?" Accessed April 15, 2021. https://www.incose.org/systems-engineering

Lego Mindstorms EV3 Help. n.d. "Using the Infrared Sensor Beacon Mode." Accessed April 22, 2021. https://ev3-help-online.api.education.lego.com/Education/en-gb/page.html?Path=editor%2FUsingSensors_Infrared_Beacon.html

Office of the Deputy Assistant Secretary of Defense for Systems Engineering. 2018. Department of Defense Digital Engineering Strategy. Washington, D.C., Office of the Deputy Assistant Secretary of Defense for Systems Engineering. https://fas.org/man/eprint/digeng-2018.pdf

Pavalkis, Saulius. 2019. "Integrating Programmable Lego Mindstorms EV3 with SysML and Cameo Systems Modeler." *Modeling Community Blog* (blog). Octoberr 1, 2019. https://blog.nomagic.com/integrating-programmable-lego-mindstorms-ev3-with-sysml-and-cameo-systems-modeler-part-1-2/

Pavalkis, Saulius, and Gan Wang. 2019. "A Model-Based V&V Test Strategy Based on Emerging System Modeling Techniques." Orlando, FL. Presented at the 29th Annual INCOSE International Symposium.

Peak, Russell, Paredis, Chris, McGinnis, Leon, Friedenthal, Sanford and Burkhart, Roger.2009. "Integrating System Design with Simulation and Analysis Using SysML." *INSIGHT:* December 2009, vol. 12, no. 4, 40– 44.

Pike, John. 2003. “MK 15 Phalanx Close-In Weapons System.” Federation of American Scientists: Military Analysis Network. Last modified January 9, 2003. https://fas.org/man/dod-101/sys/ship/weaps/mk-15.htm

Solberg, Albert. 2020. “Model Based Systems Engineering.” NASA. Last modified February 26, 2020. https://www.nasa.gov/consortium/ModelBasedSystems

Stoner, Robert H. 2009. “R2D2 with Attitude: The Story of the Phalanx Close-in Weapons.” Naval Weapons, Naval Technology, and Naval Reunions. Last Modified October 30, 2009. http://www.navweaps.com/index_tech/tech-103.php

Uppal, Rajesh. 2021. “DOD is Adopting Digital Engineering Strategy Based on Model-Based Systems Engineering (MBSE) for Military System of Systems (SOS).” International Defense, security and Technology. Last modified January 11, 2021. https://idstch.com/technology/ict/dod-is-adopting-digital-engineering-strategy-based-on-model-based-systems-engineering-mbse-for-military-system-of-systems-sos/

Visual Paradigm. n.d. “MBSE and SysML” Accessed April 28, 2021. https://www.visual-paradigm.com/guide/sysml/mbse-and-sysml/

Wilson, T Areca. 2020. “Acquisition Chief Calls for Disruptive Agility, New Digital Paradigm.” U.S. Air Force. September 16, 2020. https://www.af.mil/News/Article-Display/Article/2350344/acquisition-chief-calls-for-disruptive-agility-new-digital-paradigm/

Zwemer, Dirk. “Using Paramagic and SysML in Combination with UPDM.” Accessed 20 Feb 2021. https://www.nomagic.com/attachments/article/420/UPDM_ParaMagic.pdf

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California