3-2021

Modeling Situation Awareness in a UAV Scenario Using SysML

Tommy R. Hernandez

Recommended Citation

Hernandez, Tommy R., "Modeling Situation Awareness in a UAV Scenario Using SysML" (2021). *Theses and Dissertations*. 5039.
https://scholar.afit.edu/etd/5039

MODELING SITUATION AWARENESS IN A UAV SCENARIO USING SYSML

THESIS

Tommy R. Hernandez, Captain, USAF

AFIT-ENV-MS-21-M-234

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A.
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT-ENY-MS-21-M-234

MODELING SITUATION AWARENESS IN A UAV SCENARIO USING SYSML

THESIS

Presented to the Faculty
Department of Systems Engineering and 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

Tommy R. Hernandez, BS
Captain, USAF

March 2021

DISTRIBUTION STATEMENT A. APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED.

AFIT-ENV-MS-21-M-234

MODELING SITUATION AWARENESS IN A UAV SCENARIO USING SYSML

Tommy R. Hernandez, BS
Captain, USAF

Committee Membership:

Thomas C. Ford, PhD
Chair

Michael E. Miller, PhD
Member

Michael F. Schneider, PhD
Member

AFIT-ENV-MS-21-M-234

Abstract

The United States Department of Defense (DoD) is rapidly moving towards using systems with increasing levels of automation. Unmanned Aerial Systems (UAS) can operate with less human input than ever before and will be used in contested environments where direct human control may be impossible for long stretches of time. This increase in automation brings with it challenges in maintaining resilient Situation Awareness (SA) for operators, who still need to set goals for the UAS to accomplish their missions. Simultaneously, the DoD is increasing the adoption of Model-Based Systems Engineering (MBSE) in the development of weapon systems. Incorporating integrated digital models throughout the development process enables faster and better design. Unfortunately, until now those models have focused more on hard designs of systems and less on the socio-technical aspects. This thesis seeks to blend the use of MBSE with operator SA in automated aircraft systems. A model of a UAS is created with Systems Modeling Language (SysML) and No Magic's Cameo Systems Modeler to track the flow of information that contributes to operator SA. That system model is integrated with an environmental context and simulated through time. The results demonstrate that it is possible to measure the elements of information that are obtained from the environment and passed through the UAS to the operator. Such a model could be used to refine UAS design to enable more resilient SA in loss of communication scenarios. However, challenges remain with the use of SysML and how efficient it can be in creating the model and simulation environment necessary to be of benefit.

Acknowledgments

I would like to thank my thesis advisors, Dr. Michael Miller and Dr. Thomas Ford, for their guidance and support throughout the course of this thesis effort. Without their knowledge and experience in the Situation Awareness and SysML fields respectively, this work would not have been possible, and the time that both spent towards this endeavor is greatly appreciated. Above all, I would like to thank my wife for her patience and support throughout this whole process.

Tommy R. Hernandez

Table of Contents

Page

List of Figures

Page

List of Tables

Page

MODELING SITUATION AWARENESS IN A UAV SCENARIO USING SYSML

I. Introduction

General Issue

There is interest in the Department of Defense (DoD) in maximizing the effectiveness of human-agent teams. According to the 2018 Department of Defense Artificial Intelligence Strategy, "The United States, together with its allies and partners, must adopt AI to maintain its strategic position, prevail on future battlefields, and safeguard this order" (p. 5). Using Artificial Intelligence (AI) for defense applications, particularly for controlling unmanned vehicles, can be a force multiplier in many domains. It can be used to "to reduce risk to fielded forces and generate military advantage" (United States Department of Defense, 2018, p. 6). The strategy also "directs that we will use AI in a human-centered manner to… reduce inefficiencies from manual, laborious, data-centric tasks," in order to "shift human attention to higher-level reasoning and judgement, which remain areas in which the human role is critical" (United States Department of Defense, 2018, p. 6). However, there are still questions about how to best design human-agent systems so that the agent increases the effectiveness of the system while not negatively affecting the workload for the human. This is an active area of research. To this end, the 2016-2019 Progress Report: Advancing Artificial Intelligence R&D articulates eight national strategies for AI R&D, the second of which is "Develop effective methods for human-AI collaboration". It notes that while "[human-AI collaboration] efforts align with specific organizational missions, they also cluster around generalizable topics such as human-machine cognition, autonomy, and agency in the contexts of trusted machine intelligence, decision support, situational awareness" (p. 14).

Along with the push to maximize human-agent teams, the DoD is also investing in digital design engineering to enable a "faster, agiler, and more competitive weapons-buying process" (Roper, 2020, p. 1). Digital Engineering and Management is one of these efforts, highlighted by the creation of digital threads and digital twins. Digital threads are "extensible analytic frameworks to *connect* models – and all associated data, software and functional support – governing more than one system lifecycle phase with one-to-one real-world traceability" (Roper, 2020, p. 4). Digital twins are "one-to-one system models, cojoined with their individual real-world systems in data feedback loops, which may or may not be governed by a full digital thread" (Roper, 2020, p. 4). While there has been success in using Digital Engineering and Management for projects like the T-7A RedHawk and Ground Based Strategic Deterrent (Roper, 2020, p. 1), there is also the need to focus on the human cognitive element of system design. This is an area that requires attention, since according to Dr. Will Roper, the former Assistant Secretary of the Air Force for Acquisition, Technology, and Logistics, "Accepting analog or disconnected models will cut your digital thread, glitch conclusions, and result in acquisition-as-usual déjà vu. Completing the digital thread should be the first phase of new programs." The analog or disconnected model that he mentions could be something like a pilot-vehicle-interface study, where the results are not fed into the digital thread to influence design decisions in a cohesive manner. Many elements of human factors engineering can be included in models, such as anthropomorphic measurements and requirements. However, cognitive elements like situation awareness (SA) are not, due to the difficulty in measurement. This disconnect presents an opportunity for improvement – to develop a digital model of system SA that can be integrated into a digital thread.

Problem Statement

One of the challenges with providing SA to operators of unmanned platforms is how to provide important information from the remote platform when communications may be cut off or degraded at different points in time. Current systems like the MQ-9 do nothing to inform the operator about events that have transpired during a loss in communications. The operator is left to figure things out based on real-time data coming from the Unmanned Aerial Vehicle (UAV). This problem will become worse as higher levels of AI enable UAV use in more contested environments, where communications are expected to be degraded and cut off for large portions of the mission. Therefore, it is important to develop a system that supports SA resilience through the entire phase of a UAV mission. One way of doing this is to present an estimate of the UAV state to the operator when communications are lost, based on the last known mission of the UAV and certain environmental factors. The state estimate can also exist on the UAV. Information can be recorded on-board the UAV and compared to that estimated state. When the UAV finally reconnects with the operator, it could then upload important information to the operator based on relevance to the mission and divergence from the estimate, and that information could be used to update the estimate being displayed to the operator.

One of the main challenges with constructing a model of such an SA system is how to integrate it within a larger system model. If it cannot be integrated within a system model, the digital thread will be broken and discontinuity in design could occur.

This thesis will seek to accomplish this by developing a system model using Systems Modeling Language (SysML) which is comprised of the required structure and behavior, then creating environmental behavior associated with an operationally relevant scenario that can be integrated with the system model to simulate and assess SA repair capability in certain contexts. Using SysML, an established open-source specification which is already used throughout the DoD, enables this work to flow into existing system models. The other challenge with this approach is how to synchronize environmental behavior with system behavior in such a controllable way that allows operationally relevant scenarios to be simulated in SysML. This model will serve as a test case to determine the efficacy of using SysML with the software tool Cameo Systems Modeler to develop not only the SA repair system, but also the simulation environment necessary to analyze its effectiveness.

Research Objectives

1. Create a model of an SA repair system in a UAV, including all relevant system blocks, behavior, and information elements, and determine how to integrate that with environmental blocks and behavior.
2. Determine how effective and efficient SysML and Cameo Systems Modeler can be for creating a discrete event simulation environment that models an operationally relevant scenario.

Research Questions

1. How can SysML and Cameo Systems Modeler be used to simulate a scenario where information is passed from the environment, through a UAV, and to an operator?

    a. What mechanisms should be used to keep the system behavior synchronized with the environmental behavior?
    b. How should information elements be stored and passed throughout different sub-systems of the model?
    c. How can information elements be estimated after communication loss between the UAV and operator control station (OCS)
2. In what ways can the results of the simulation be useful for analysis to improve the SA repair capability of the system design?

Assumptions/Limitations

The scope of this thesis covers a system that generates and passes two simple information elements. It is meant to be a proof of concept that tests the viability of using SysML to create this model and simulation. Therefore, two information elements are chosen to convey different types of data - events from the environment, and relatively steady state information from the system itself (which is still influenced by the environment). Additionally, the model receives no input from the operator. The focus is on sending the right information to the operator to support SA. Although operator intent and commands play a large part in how a situation is understood, including that dynamic is beyond the scope of this thesis.

II. A Methodology for Modeling Situation Awareness in a UAV Scenario Using SysML

Tommy R. Hernandez
Air Force Institute of Technology
Wright Patterson AFB, Ohio

Abstract

The United States Department of Defense is rapidly adopting Model-Based Systems Engineering to improve acquisition agility. These methods permit the early-stage evaluation of multiple system alternatives. However, current methods focus predominantly on the development of hard systems and provide little insight into the impact of these systems on the larger socio-technical system. Further research is required to reflect the representation of humans within these systems. The current research attempts to model human situation awareness (SA) for Unmanned Aerial Vehicles (UAVs) having increasingly autonomous capabilities which communicate with a remote operator through an unreliable communications channel. The method focuses on modeling Level 1 SA where information elements are passed from the environment to an operator control station and perceived by the operator. Without the perception of information elements, the ability to create more accurate Level 2 and 3 SA assessments is degraded. Thus, the goal of the model is to understand how different UAV communications systems architectures influence Level 1 SA. In our approach, components of the system which are important to maintaining SA are modeled in the context of specific scenarios. The importance of information elements within these scenarios is understood through goal-directed task analysis and operator assessment during system simulation. Systems are evaluated based on their ability to pass important elements of information to the operator. Although this modeling method permits the estimation of Level 1 SA for a given scenario, application of this method requires customization to each scenario of interest. Future research will explore extending this method to higher levels of SA.

Keywords

Model Based Systems Engineering (MBSE), Situation Awareness (SA), autonomous systems

1. Introduction

The Department of Defense (DoD) is interested in maximizing the effectiveness of human-agent teams (HATs), which includes the adoption of Artificial Intelligence (AI) to maintain its strategic position, permitting it to succeed on future battlefields [1]. Using Artificial Intelligence (AI), particularly for controlling unmanned vehicles, can be a force multiplier in many domains. In addition, the adoption of AI in HATs can reduce inefficiencies from laborious tasks and shift human attention to higher-level reasoning and judgement to maintain system resilience [1]. However, there are still questions about how to best design HATs so that the agent increases the effectiveness of the system without negatively affecting human situation awareness and workload. Thus, recent guidance lists “Develop effective methods for human-AI collaboration” as the second of eight national strategies for advancing AI research and development. This document specifically lists “human-machine cognition, autonomy, and agency in the contexts of trusted machine intelligence, decision support, situational awareness” [2]. Loss of Situation Awareness (SA) can lead to the out-of-the-loop problem described by Endsley, where people can be slow to detect automation problems and correct them, reducing the effectiveness of the system [3].

SA is defined as “the perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the future” [4]. It reflects the current state of knowledge, not the process used to obtain the knowledge [4]. According to the model that Dr. Endsley created, a person’s perception of elements forms the basis of their SA, which affects their actions and performance, along with other factors like doctrine and training [4]. Exploring the model further, Level 1 SA is the Perception of the Elements in the Environment [4]. Level 2 SA goes further than perception as the operator should understand “the significance of those elements in light of pertinent operator goals” [4]. Level 3 SA builds upon Level 2 and is “the ability to project the future actions of the elements in the environment” [4].

Many attempts have been made to quantify and measure SA. Some methods study process and performance metrics using techniques like eye tracking. These methods seek to identify perceived knowledge, then attempt to understand how that knowledge guides future operator actions, including eye movements, to gather information to confirm or refute hypotheses supporting associated goals. As Endsley suggests, this method provides insight into how people develop SA but are limited in assessing quality and completeness of SA [5]. One technique is the Situation Awareness Global Assessment Technique (SAGAT). This technique uses simulations of tasks, which are frozen at randomly selected times during which the system displays are blanked and operators are queried about their current perception of the situation [5]. The questions are determined from the operator's SA requirements, which are derived from the results of an SA requirements analysis, such as the Goal Directed Task Analysis [5]. The operator's answers about their perceived situation are then compared to the real situation to provide an objective measure of SA. SAGAT scores are usually expressed as percent correct for each question [5]. There are other scoring variants that combine the scores from all questions, combine them into level 1, 2, and 3 SA, or other more specific domain type variants. SAGAT scores have been shown to predict operator performance and decision-making measures.

SAGAT and similar methods allow an individual's SA to be assessed while performing tasks within a simulated or realistic environment. However, it requires a human-in-the-loop study, which are scenario specific and require significant time and other resources to execute. SA assessment using analytic models has not been researched extensively. One attempt differentiated tactical and strategic SA in tactical airlift missions [6]. This effort linked tactical SA to specific tasks such as looking at a head-up display or a moving map display. Strategic SA was not linked to a specific task but was assumed to increase if operators were not task saturated on observable and cognitive tasks associated with near term goals. Once workload increased to a level which indicated task saturation, strategic SA was assumed to decay. Through these assumptions and a model of operator workload, the study estimated the levels of tactical and strategic SA during simulated missions. The results of this model indicated that the automation present in a new platform largely compensated for the loss of crew in an old platform when it came to individual SA (mainly the pilot and co-pilot), although total team SA was slightly reduced. In this model SA was dissociated from workload since strategic SA decayed when the crew were overloaded and near-term tasks increased tactical SA.

To reduce the time required for system acquisition, the United States Air Force has recently embarked on a mission to adopt Model-Based Systems Engineering (MBSE) [7]. The envisioned efficiencies of this process change stem from the reuse of information, increasing the sharing and understanding of design artifacts, and speeding and enhancing robust decision-making. While models of technical components are well understood, the integration of these models with models of human behavior and performance requires further development. Thus, our research attempts to understand how one might construct generalizable models of human SA for use in MBSE analysis of systems employing HATs.

2. Understanding the Relationship Between Information Elements and SA

Model development requires the definition of a quantitative output variable, taking inspiration from Endsley's SAGAT technique, the output variable for Level 1 SA in our model is the proportion of SA-relevant information elements conveyed. Information elements have been listed in the literature for aircraft [2] and for UAVs [8] as shown in Table 1. Endsley lists these SA elements by level while Drury and colleagues decomposed SA elements into ten categories.

Table 1 Endsley's (1) and Drury's (2) Information Elements by SA level and source.

Source	**Level 1**	**Level 2**	**Level 3**
Environment	location[1], altitude[1], and heading of other aircraft[1]; location of ground threats and obstacles[1]; Weather near the UAV[2]	3D spatial relationships[2]; Operational Threats[2]	predicted 3D spatial relationships[2]
Mission	current target[1]; UAV's mission[2]	mission timing and status[1]; impact of system degradation[1]; tactical status of threat aircraft[1]; UAV's progress towards completing the mission[2]	projected aircraft tactics and maneuvers[1]; firing position and timing[1]
UAV	location[1], altitude[1], and heading of ownship[1]; system status; Health of the UAV[2]; Status of the UAV[2]	time and distance available on fuel[1]; 3D spatial relationships[2]; Logic of the UAV[2]	predicted 3D spatial relationships[2]; Degree to which UAV can be trusted[2]

Comm Subsystem	system status[1]		
Operator Workstation	system status[1]		

In our example, the UAV system is composed of the UAV, a communication subsystem, and an operator workstation. However, information elements can also originate from the environment or from mission goals. Thus, besides categorizing the information elements from these sources by SA Level, Table 1 also lists the proposed source of each of these information elements. In our current model we also make the simplifying assumption that SA loss occurs due to system deficiencies and perfect transfer of information occurs from the operator workstation to the human enabling all higher levels of SA.

3. SA Repair System Comparisons

Systems that are operated remotely, like UAVs, can severely degrade SA during temporary communication system failure. This can result in the loss of Level 1 SA, which degrades the ability to create more accurate Level 2 and Level 3 SA assessments. Future UAVs may continue limited operations autonomously during communication loss but regaining operator SA once communications are reestablished remains an issue. Such a UAV will likely be responsive to high level commands provided through methods such play calling [10]. The importance for the current research is to assess methods to permit the UAV tasks and environmental changes to be clearly and concisely communicated to the operator in an expeditious manner. To develop our model, we began by envisioning system alternatives to model. These alternatives are depicted in Figures 1, 2, and 3.

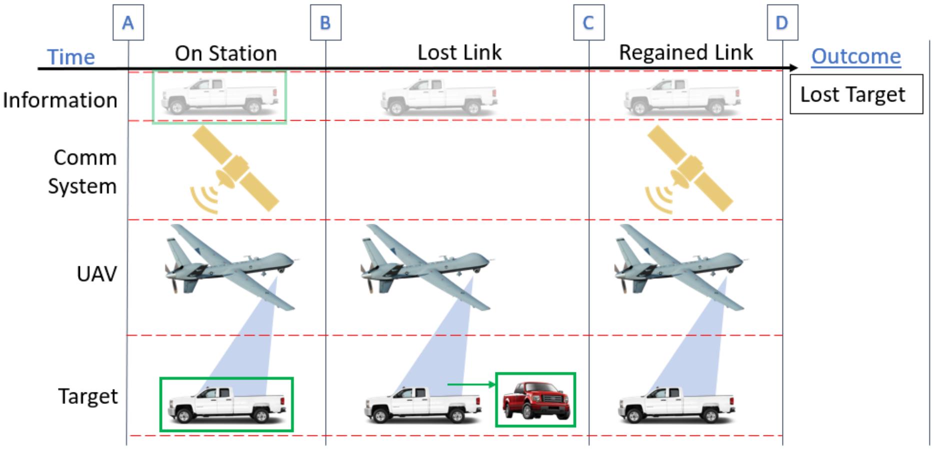

Figure 1 UAV System behavior associated with System Alternative 1

As shown in Figure 1, the first alternative is a UAV capable of autonomously tracking a target during lost communications using onboard automation. However, all communication is lost between the UAV and the operator. See Figure 1 for a detailed illustration of the scenario. As shown between times A and B, the UAV is tracking a Potential Kidnap Victim (PKV) in a white pickup truck where the green box denotes the location of the PKV. From time B to C, communication is lost and the PKV switches vehicles from the white truck to a red truck. However, the automation on the UAV is tracking the white truck and remains fixed on it. When communication is re-gained from time C to D, the operator has no SA of the vehicle switch. The outcome of this mission is a lost target.

One solution overcoming this loss of SA is for the UAV to record all information it encounters like a Digital Video Recorder (DVR). As illustrated in Figure 2, the operator can watch the recording upon completion of the mission. However, such a late repair of SA is of limited utility in many situations. As shown, the operators can determine that the vehicle switch occurred. Unfortunately, the target is still lost, but some information is known about the new vehicle and perhaps an initial direction of travel.

The third system alternative is shown in Figure 3. As shown, when communications are lost between B and C, the UAV records the activities within its field of view. When the link is restored during time interval from C to D, the system attempts to download both the recorded information as well as provide real-time communications. As the bandwidth of the system is constrained, however, this alternative will also need a method of prioritizing information transmission. In an ideal system, the prioritized set of information from the UAV would include information that passengers disembarked the white truck and entered the red truck, potentially enabling the operator to re-gain the target and continue the mission.

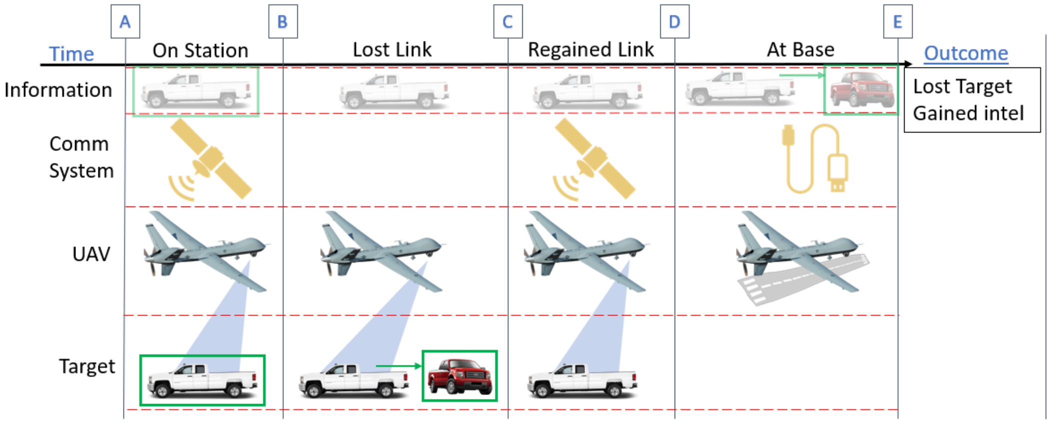

Figure 2 UAV System behavior associated with System Alternative 2, including data recording.

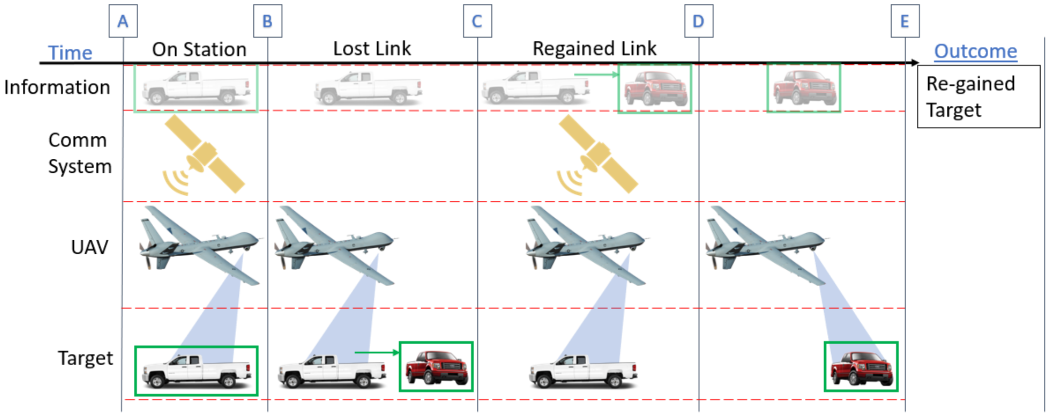

Figure 3 UAV System behavior associated with System Alternative 3, including information prioritization.

To aid in the rapid repair of SA, a system is needed to not only record the activities and sensor data of the UAV during lost communications, but to prioritize information transmission to the operator upon regaining communications. Efficiency is important, since bandwidth will be limited and it would take the operator time to review information while continuing the mission. Different system architectures and approaches to prioritizing and conveying this information must be considered. Ideally, our model of operator SA would permit an SA estimate for each system alternative as it is developed.

In our proposed model, elements of information will be modeled as objects that can be perceived and calculated by the UAV and passed to the Operator Control Station (OCS). The relative importance of those elements at any point in any given scenario will be determined by interviewing subject matter experts (SMEs) about what they think would be important at the different points during the scenario, like the approach Endsley suggests for SAGAT. During the simulated mission, communication loss will be modeled at different points in time stress the different system alternatives. Additionally, intermittent communication will be modeled to stress the prioritization capability of the system, to ensure important elements are transmitted given bandwidth constraints.

4. Proposed Modeling Methodology

The proposed methodology is presented as a Systems Modeling Language (SysML) activity diagram in Figure 4. The first step is to decompose the information elements and group these elements into categories based on whether they predominantly affect each subsystem, the environment, or the mission as illustrated in Table 1. This is important when modeling how the elements pass among those system elements. These information elements are then decomposed further into the individual information elements that can be passed from the UAV to the OCS and the human operator. An example of this is to start at Operational Threat, which can be decomposed to Threat Identification, Vehicle Type, and Vehicle Image. An operator can receive an information element like an image of a person, or a vehicle identification, and use that information to increase their SA.

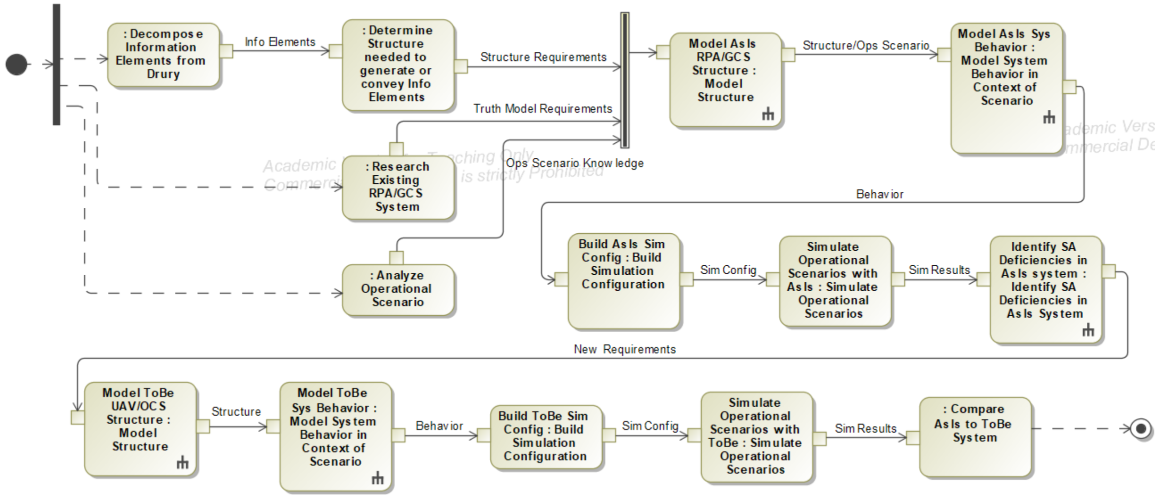

Figure 4 Methodology Activity Diagram

Once the information elements are decomposed, the next three steps occur in parallel. They include determining the structure needed to generate or convey the information elements and researching the existing system. These steps are required to model the structure of the existing system. It is necessary to know what structure is needed to generate or convey the information elements because there are many parts of the existing system that are unnecessary to model if the area of interest is only the SA of the operator. The Operational Scenario provides the context in which the system behavior will be analyzed. That context is relevant to the priority of the information elements, which affects the necessary structure. For example, if the scenario does not include geographic obstruction, topographic data would not be needed by the operator, and the structure required to generate and convey that data would not be needed.

The prime objective of the model is to quantitatively describe the information elements passed to the operator. Depending on the automation, recording, and prioritization capabilities of the system, different amounts of information representing different information elements will be passed. This will enable a baseline comparison between the as-is system (i.e., Alternative 1) and any to-be system (Alternatives 2 and 3). The second objective is to qualitatively assess SA utilizing UAV operator feedback. This project will use videos gathered from a previous UAV simulation. The video will allow operator SMEs to determine how much SA value any given information element has at any given point in the simulation. The next step is modeling the structure of the as-is UAV/OCS. This involves creating a high-level physical decomposition of the required structure, consisting of things like the UAV, OCS, and Communication system. Next, key internal and external interfaces are identified and modeled, as well as the lower-level Communication, Command, and Control (C3) components. The components modeled here are dependent upon the analysis performed in the previous steps.

The next step is to model the as-is system behavior within in the context of the operational scenario. For better flexibility and modularity, this step applies both state and activity diagrams. The environment is decomposed into states that correspond to different phases of the scenario. For example, the environment could have different weather states corresponding to different levels of visibility, or different electromagnetic conditions corresponding to the presence or absence of jamming. Different activities could occur within those states, such as enemy movement only happening in the presence of cloud cover. The system states and activities are modeled to correspond with what is happening in the environment during the scenario. An example of a system state is having adequate fuel for the mission, or only enough fuel to return to base. The activities within those states might include continuing the mission as normal or returning to base. The communication system behavior is also modeled and will have states and activities, such as a good-link or no-link state. After those three aspects of behavior are modeled, the information elements being passed are modeled. These elements have associated parameters like bandwidth requirements. For example, a track file of a vehicle is composed of less data than an image of a vehicle. These information elements will be either passed through or filtered by the communication system. The filtering will be dependent upon the size of the information element, status of the system and environment, and importance and relevance of the information, as determined by the system.

After the behavior is modeled, a simulation configuration is created and executed. The configuration determines what parameters are recorded and where they are stored in the model. The results of that simulation are then analyzed to determine what deficiencies are present in the as-is system. If information elements useful for SA are not collected or passed to the operator, the ratio of the required information elements passed will be used as a measure and individual elements will provide areas for improvement of a future to-be system. New requirements for structure and behavior can then be designed to generate and convey those elements to the operator. The new requirements are used to create the to-be system, which follows the same process as the creation of the as-is system. The last step in the process is to compare the as-is system to the to-be system and examine relevant SA improvements.

5. Discussion and Conclusion

The next steps in the project are to create the model and run the simulation. The information elements passed to the operator will be cataloged and examined. Further improvements can then be made to increase the amount of information elements that are passed. After that, SME feedback will be solicited about what information is important at different times in the scenario. The biggest challenge with this approach is how sensitive any SA analysis is to the operational context. The model can be created and analyzed for a specific scenario, but it is difficult to predict its applicability to any other scenario. That makes prioritizing certain information elements difficult. A way to mitigate this challenge would be to run the model in many scenarios and see if there are some information elements that tend to be important across a wide variety of similar scenarios. For example, an information element like target location could turn out to be important across most tracking missions, whereas friendly location could turn out to be more important in specialized missions. Prioritization schemes could then be created that adapt to the mission type.

The above methodology will allow the modeling of SA in response to alternative UAV system designs in the face of communications failure. Developing and utilizing an automated UAV capable of executing missions in communication-denied environments is consistent with the 2018 Department of Defense Artificial Intelligence Strategy and the modeling approach supports future model-based systems initiatives. The resulting model will enable Level 1 SA to be quantified in terms of information elements received by the operator, which is the basis for achieving Level 2 and 3 SA [5]. This will ultimately enable SA analysis using MBSE.

References

[1] United States Department of Defense. (2018). Summary of the 2018 Department of Defense Artificial Intelligence Strategy. 1–17. https://media.defense.gov/2019/Feb/12/2002088963/-1/-1/1/SUMMARY-OF-DOD-AI-STRATEGY.PDF

[2] Artificial Intelligence Research & Development Interagency Working Group, et. al. (2019). 2016-2019 Progress Report: Advancing Artificial Intelligence R&D. 1-40. https://www.nitrd.gov/groups/AI

[3] Endsley, M. R. (2017). From Here to Autonomy: Lessons Learned from Human-Automation Research. Human Factors, 59(1), 5–27. https://doi.org/10.1177/0018720816681350

[4] Endsley, M. R. (1995). Toward a theory of situation awareness in dynamic systems. Human Factors, 37(1), 32–64. https://doi.org/10.1518/001872095779049543

[5] Endsley, M. R. (2019). A Systematic Review and Meta-Analysis of Direct Objective Measures of Situation Awareness: A Comparison of SAGAT and SPAM. Human Factors. https://doi.org/10.1177/0018720819875376

[6] Meyer, D. (2015). Effects of Automation on Aircrew Workload and Situation Awareness ... DTIC Science & Technology. https://apps.dtic.mil/dtic/tr/fulltext/u2/1003569.pdf

[7] Roper, W. (2020). Take the Red Pill: The New Digital Acquisition Reality.

[8] Drury, J. L., Riek, L., & Rackliffe, N. (2006). A decomposition of UAV-related situation awareness. HRI 2006: Proceedings of the 2006 ACM Conference on Human-Robot Interaction, 2006, 88–94. https://doi.org/10.1145/1121241.1121258

[9] Endsley, M. R., Bolte, B., & Jones, D. (2003). Designing for Situation Awareness. In Designing for Situation Awareness. https://doi.org/10.1201/

[10] Miller, C. A., & Parasuraman, R. (2007). Designing for Flexible Interaction between Humans and Automation: Delegation Interfaces for Supervisory Control. Human Factors, 49(1), 57–75. https://doi.org/10.1518/001872007779598037

III. Using SysML for Discrete Event Simulation of a Situation Awareness Repair System for a UAV

Tommy R. Hernandez

Air Force Institute of Technology

Wright Patterson AFB, Ohio

Abstract

The United States Department of Defense is rapidly adopting Model-Based Systems Engineering to improve acquisition agility. These methods permit the early-stage evaluation of multiple system alternatives. However, current methods focus predominantly on the development of hard systems and provide little insight into the impact of these systems on the larger socio-technical system. Further research is required to reflect the representation of humans within these systems. The current research attempts to model human situation awareness (SA) for Unmanned Aerial Vehicles (UAV) having increasingly autonomous capabilities which communicate with a remote operator through an unreliable communications channel. The method focuses on modeling Level 1 SA where information elements are observed from the environment and passed to an operator control station and perceived by the operator. Without the perception of information elements, no higher level of SA can be achieved. Thus, the goal of the model is to understand how different UAV communications systems architectures influence Level 1 SA. A proof-of-concept model is created with a subset of two information elements using the Systems Modeling Language (SysML). Lessons are learned about the efficacy of using SysML and Cameo Systems Modeler. Future research will explore extending this method to higher levels of SA and using the simulation to improve system design.

Keywords

Model Based Systems Engineering (MBSE), Situation Awareness (SA), autonomous systems

1. Introduction

Challenges in the application of the method described in the previous section require further exploration. The fundamental idea is to create a model of an Unmanned Aerial System (UAS) system that can provide resilient situation awareness (SA) during and after communication loss with the operator, then simulate the model to see how well it accomplishes that task. The larger goal is to develop a legacy UAS model with real-time control and feedback and new UAS with SA repair capability that passes a full set of information elements to the operator, then compare how many important information elements are passed with both systems. The idea behind creating the model using SysML is that is enables a well-fleshed out structure that can also have behavior representative of an operationally representative scenario. The theoretical benefit of such an approach is that it allows the creation of system blocks with specific properties and behaviors to be tested in different scenarios depending on what you want to observe. The main challenge revolves around using the modeling language SysML and the software tool Cameo

Systems Modeler by No Magic, to build a discrete event simulation environment where system behavior can be simulated through time and be synchronized with environmental behavior. The mechanism used to keep the system and environmental activities in sync is a model-based clock, where system blocks execute their behavior within a certain time increment. This allows fine grained control over how events in the environment interact with the system, which is useful for ensuring relevant information elements are passed to the operator. The first step to the larger goal is to create a prototype model that passes a smaller subset of information elements to an operator, to be able to test the modeling language and tools to determine how best to use them and what the challenges and benefits are.

2. Creating the System Structure

The structure of the prototype system is represented by a SysML block and value properties, along with their requisite relationships. At the highest level it is represented by a block called the Analysis Context. This consists of a UAV, an operator control station (OCS), the environment where the system operates, and a clock to control the timing of the simulation. The larger parts are further broken down into smaller subsystems using the composition relationship. Using the composition relationship gives part properties of the subsystems to each parent system, and reference properties of the parent system to each subsystem. These are important in the model behavior, where those part and reference properties are used both to send signals and read value properties. This is explained more in section 4. The only subsystems being modeled are those relevant to the information elements being passed. The UAV consists of the fuel system, communication system, and sensor package, which are then further broken down as shown in Figure 5. All pieces of the system and environment are joined together in a block called the Analysis Context. This allows everything to be simulated by having the Analysis Context be the target of the simulation. It is also connected to the model-based clock and allows the discrete event simulation behavior necessary to keep everything synchronized. See Table 2 for the break down of the sub systems and their general functionality.

Table 2 Sub-system breakout

I. Analysis Context	Contains the system, environment, and clock, and is the execution target of the simulation.
1. Prototype UAV	The aircraft component of the system.
1.1 Prototype Communication System	A sub-component of the UAV, relays information to the OCS.
1.1.1 Divergence Detector	Detects divergence between the truth data and estimated data.
1.1.2 Information Recorder	Records information when UAV is disconnected from OCS.
1.1.3 System State Estimator (SSE)	Estimates the UAV system state during communication disconnect. Also estimates target location when the sensor loses sight of it.
1.2 Prototype Fuel System	Calculates the fuel quantity for the UAV.
1.3 Sensor Package	The high-level block containing the UAV's sensors.
1.3.1 Wide Area Surveillance Sensor	A sensor with fixed field of view that tracks the target.
2. Operator Control Station (OCS)	Receives information from the UAV and displays it to the operator.

2.1 Operator System State Estimator	Estimates the UAV system state during communication disconnect. Identical estimation to the UAV's SSE.
3. Environment	The high-level block containing the target and wind blocks
3.1 Target	An enemy target that moves around, and is sensed by the UAV
3.2 Wind	Wind in the environment that changes speed during the simulation, affecting fuel burn rate
4. Clock	Contains the model-based clock activity

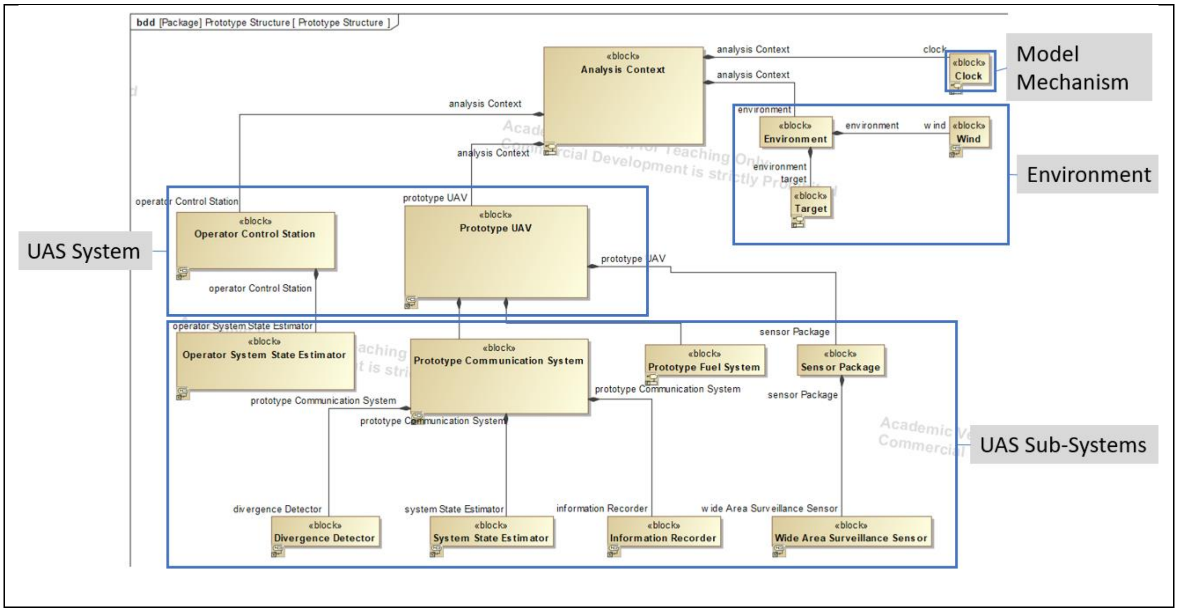

Figure 5 System Structure

The Fuel System and Wide Area Surveillance Sensor (WASS) are required because those are the system pieces which will generate the information elements to be passed to the operator. Fuel was chosen because the consumption of fuel is one of the simplest things an aircraft does, but there is some variability in burn rate due to wind that could cause a loss of SA should the operator not be aware of environmental conditions during a loss of communication. The function of the WASS is to keep track of a target over a large field of view, and relay those coordinates to the operator. This allows event driven information to be captured and sent to the operator, such as when the target leaves the field of view or comes back into the field of view. This type of sensor also keeps the system behavior simple since the sensor is not slewing to different positions. The idea is that it watches a fixed area, a town for example, and can track targets within that area.

The System State Estimators (SSEs) for the UAV and OCS are responsible for estimating information elements when truth data is not available. This occurs during communication loss with the operator, although this model also uses them to estimate target location outside of the field of view of the WASS. They use the exact same estimation techniques for each information element, so that when communication loss occurs, the UAV SSE knows what the operator is seeing from the OCS SSE. This allows the divergence to be calculated from the estimated data to the truth data, which is done at the Divergence Detector on the UAV. This divergence is used to queue up messages to be sent back to the OCS when communications are restored. Therefore, instead of only getting the immediate real-time feed of information, the operator is presented with information about how much the estimate diverged from reality, and when that occurred. This functionality lies at the heart of the SA repair capability that is being modeled.

Each system also has a set of value properties, some of which are features of the system being modeled, or are the information elements of interest, and some of which are used as variables and counters to make the behavior models work correctly. Examples of system features are things like the “current fuel quantity” and “burn rate,” which describe physical things in the fuel system. The “last fuel quantity” is an example of a value property that is a variable needed for the calculation of “current fuel quantity” in the model. Similarly, in the SSE, “estimated fuel quantity” and “estimated target location” are features of the system, however, “SSEDisconnectReceive” is a counter variable to make the modeled behavior work correctly and is not a representation of a system feature.

The environment is also represented using blocks and value properties. The relevant environmental pieces chosen for this model are the wind and the target being tracked. The wind is necessary to enable a changing fuel burn rate that can be estimated but also will diverge from truth, so the SSE and Divergence Detector can be appropriately tested. The target also has its own value properties, such as location and current vector.

3. Keeping Time in the Simulation

To understand how the behavior of the system is modeled, it is important to understand how the model-based clock works. This is a structure that is adapted from a Model-Based Clock shown on the No Magic Cameo Simulation Toolkit Documentation site [1]. There are activities within the Analysis Context and the Clock, called “SysTick” and “tick,” respectively. “SysTick” sends a “tick” signal to the clock, which then receives it in the “tick” activity, increments the “time” value property by some step value (in this case one), and sends a "next" signal to the Analysis Context. See Figure 6 for an illustration of these activities.

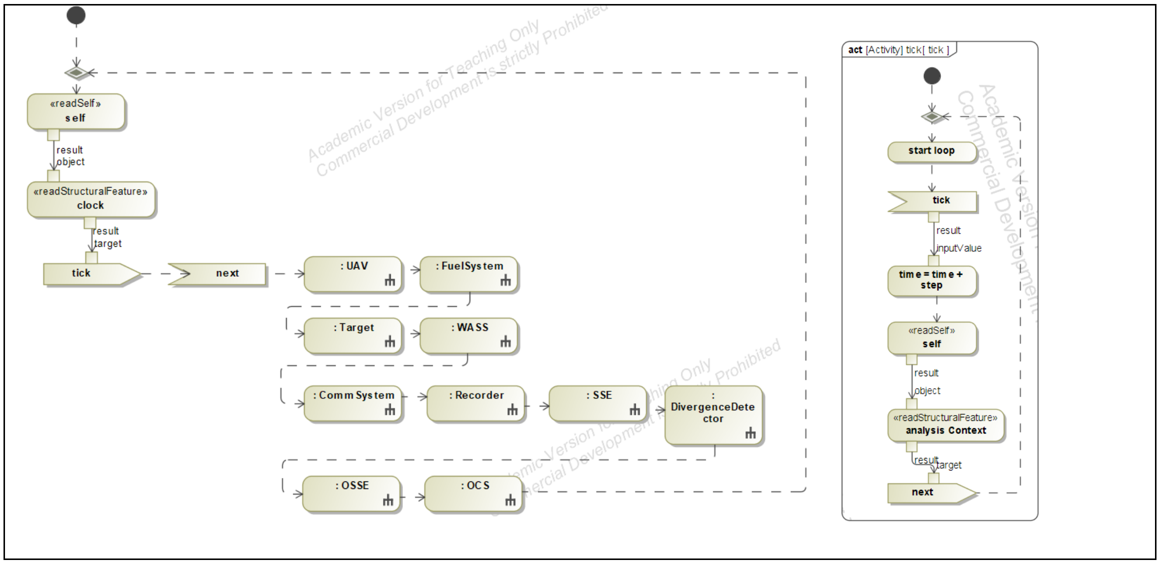

Figure 6 "SysTick" activity on the left, "tick" activity on the right

In the Analysis Context activity "SysTick", "tick" signals are also sent to each system and environmental block. Each of those system and environment blocks then contain their own "tick" receive signal actions, with which they receive the signal, carry out all of their behavior during that time increment, and end with a "next" send signal action that sends the signal back to the Analysis Context to continue down the chain of behaviors. The "SysTick" send signal action order determines system activity order of execution. It is possible some system activities could occur in parallel, but in this instance that is not required, so all activities occur in series for simplicity. The last activities to execute are those belonging to the Operator Control Station, since that is the end of the line where all information has been gathered, calculated, or estimated, and is finally displayed to the operator. Figure 7 shows the “tick” send signal action and “next” receive signal actions for the UAV system as an example. Notice there are four “UAVNext” receive signal actions – this is done to mitigate a software bug in Cameo Simulation Toolkit where occasionally a signal reception is missed and causes the model to stall [2]. Given the number of times the entire sequence is run during an execution of the simulation, four signal reception actions is enough to avoid that bug. Each of the systems with behavior that must occur within the time sequence have similar actions. As seen in Figure 6, after the UAV is the Fuel System, Target, WASS, etc. The Fuel System activity can be seen in Figure 8, where there are tick reception and next sending actions that move the control flow to the next system in line from Figure 6.

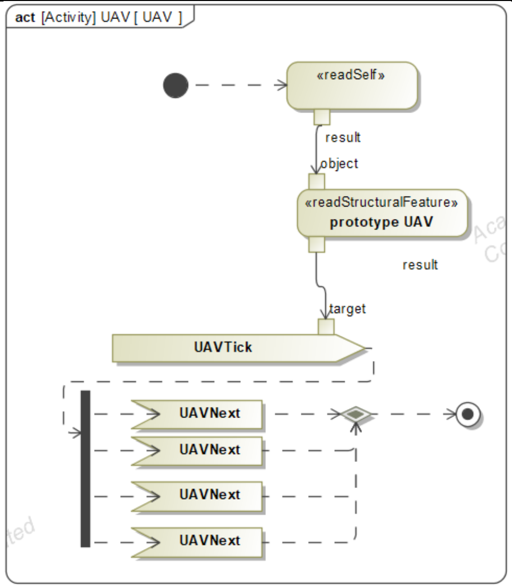

Figure 7 UAV clock Activity Diagram

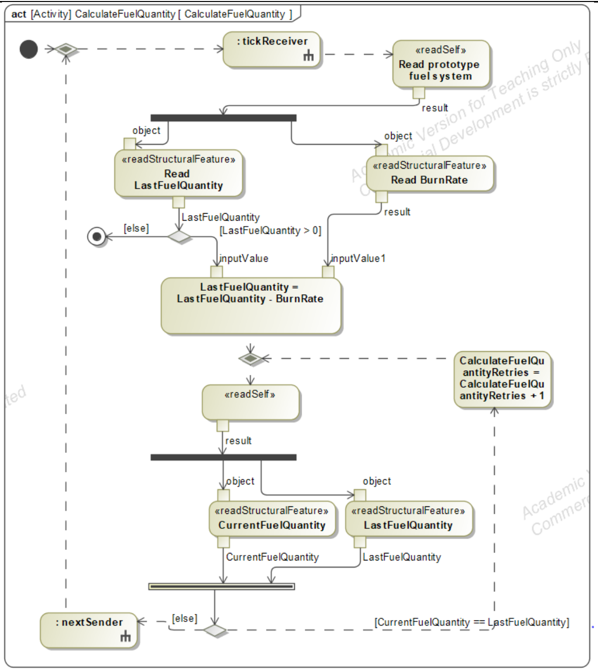

Figure 8 Calculate Fuel Quantity Activity Diagram

4. Modeling the Environment and UAV System Behavior

Knowing what the system structure looks like, and how the behavior is run with a model-based clock, it is now possible to look at the system behavior. Starting after the clock increments time, the “tick” signal is sent to the UAV block. The UAV in this model has two states, a loiter profile and a return to base (RTB) profile. While loitering, the UAV constantly checks the fuel quantity during each time increment, then sends the “UAVNext” signal to move on to the next system’s activities. Upon reaching a predetermined fuel quantity, it sends a bingo fuel warning that triggers it to transition into the RTB profile. While in that state, it flies until the fuel depletes, and the simulation ends.

The Fuel System executes its behavior after the UAV. This is where the fuel quantity calculation occurs, which is based on a base burn rate plus an additional amount based on the wind speed. The wind speed is generated base on a random function, to allow for run-to-run variability. The wind speed stays constant for a certain random amount of time, after which it changes again. The floor of the wind speed is also increased after comm loss occurs, to inject an unexpected high fuel burn rate that must be reported to the operator after reconnection. The value property “CurrentFuelQuantity,” which is owned by the Fuel System, is tied to the constraint “CalculateFuelQuantity” through a parametric diagram. The fuel calculation occurs in a Matlab function, through its integration with Cameo Systems Modeler (It was found that using Matlab in this way, even for a simple calculation, was more reliable and resulted in less issues than using the built-in math of Cameo). The activity “CalculateFuelQuantity” (See Figure 4) changes the “LastFuelQuantity” each time increment then the Matlab function uses the value property “LastFuelQuantity” and “BurnRate” to calculate the “CurrentFuelQuantity”.

After the Fuel System is the Target behavior. “Move” is the only activity the Target does during the simulation. The “Move” activity changes the Target’s vector and location. That location is relative to a center point of the sensor field of view and does not use a real-world absolute coordinate system. The Target proceeds along a certain vector for a random but bounded length of time, and that vector is bounded to certain directions for a certain length of time. The directions are one the four quarters of a compass, from 315º to 45º, from 45º to 135º, from 135º to 225º, and from 225º to 315º. These directions are given a 25% probability of occurrence every time the direction changes. The vectors are bounded in directions like this to make it more realistic, where a target is more likely to be going to a certain destination, and less likely to be randomly bouncing around a central point. This also allows for better target location estimation when the target leaves the field of view of the sensor.

The WASS behavior is after the Target’s behavior. The goal of the WASS is to read the Target’s location and send that information to the Communication System where it can be sent to the OCS. The mechanism for doing this is to receive signals from the Target, which carry the target location and direction parameters. Those values are then stored in value properties owned by the WASS, which are then able to be read by the Communication System. An interesting thing to take note of is that there are two ways values are transferred to different systems in this model. One way is to send signals from their source to a destination. The other is to read those value properties from the destination using a series of “read self” and “read structural feature” to the source location. The “readStructuralFeature” action reads part properties or reference properties all the way until it reads the value property in question. An example of both methods can be seen on the left side of Figure 9, where the Target location and travel direction signals are received, and their parameters are passed into “addStructuralFeatureValue” actions. On the right side is the series of read actions which end in the Target’s location radius being passed to a guard, where the WASS determines whether the target is in the sensor’s field of view. Both methods of reading values are used to explore the efficacy and challenges of each.

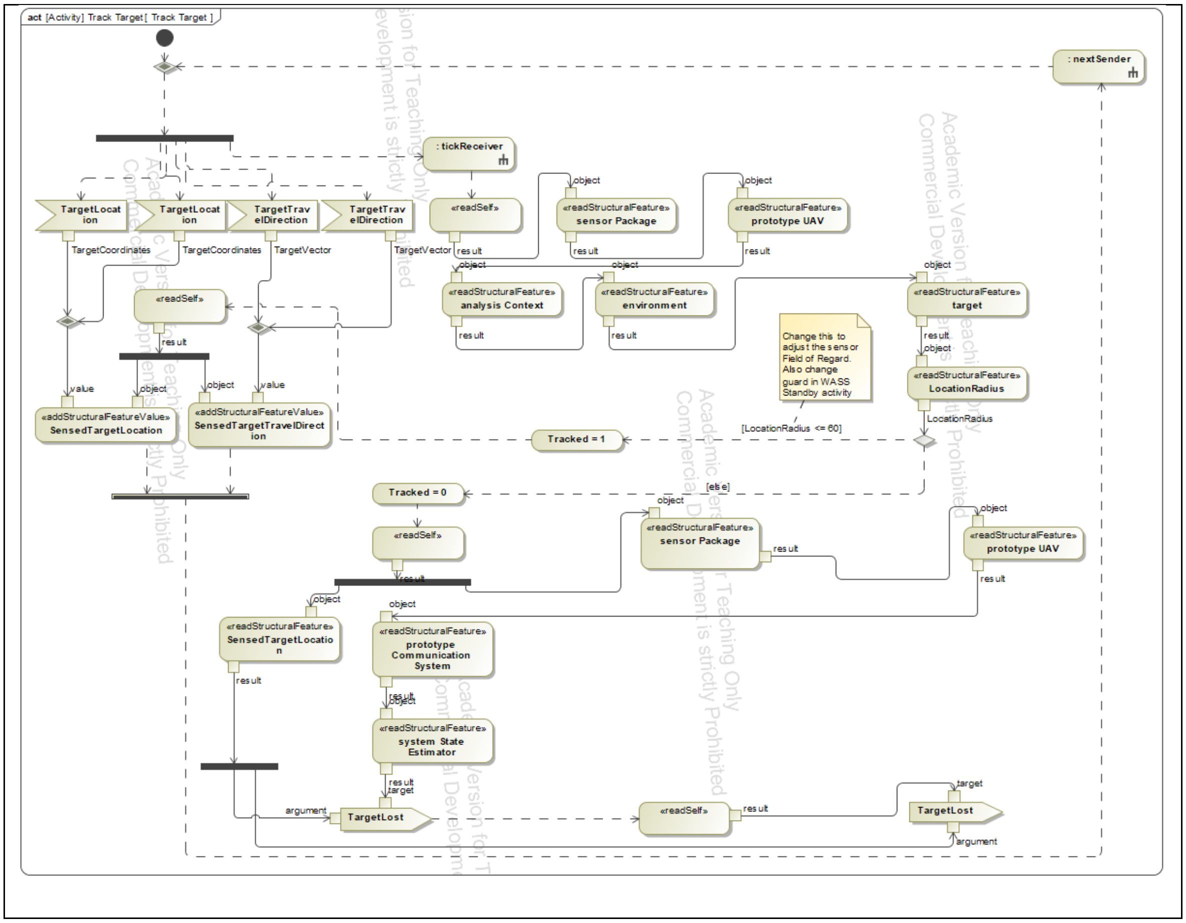

Figure 9 Track Target Activity Diagram

The Communication System behavior is in many ways the nexus of the entire system. This is where the sensor and fuel information meet to be sent to the OCS. This is the second system mentioned, along with the UAV, to have multiple states, represented in a state machine diagram. The three states can be seen in Figure 10. The Initial Pass Information Activity is done during the Connected state. This activity takes the information generated from the Fuel System and WASS, and sends it to the OCS by way of send signal actions. The Communication System contains a value property called "CommTime", which is linked to the "time" value property of the Clock block in a parametric diagram. At time 50, the communication link is disconnected, and the system transitions to the Disconnected state. During this state, the system enters the "Standby" activity. During "Standby", it is waiting for three different possible messages that could be queued up to send to the OCS after reconnecting – a target lost warning, a target divergence warning, and a fuel divergence warning. The target lost warning comes from the WASS, if the target location goes outside of the field of view of the sensor. The target divergence and fuel divergence warnings come from the Divergence Detector, if either estimated value goes outside of a set tolerance that is entered before the beginning of the simulation. When the Communication System reconnects with the OCS at a certain time, any queued warnings are sent followed immediately by the real-time data generated by the WASS and Fuel System.

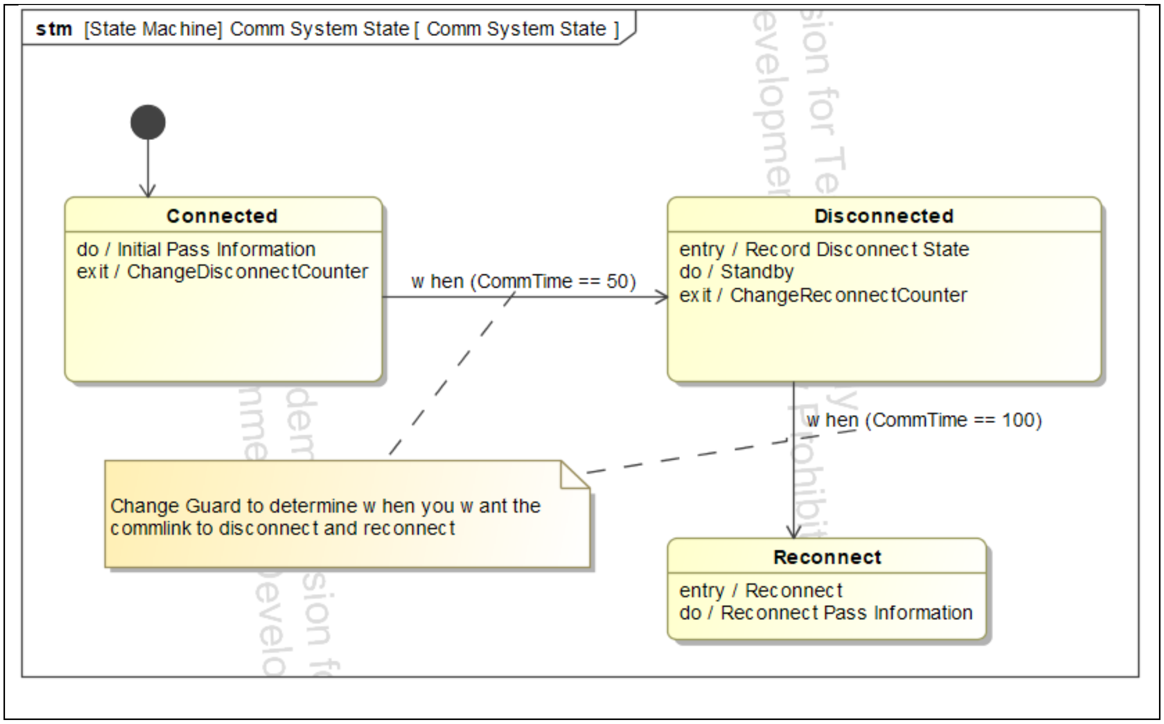

Figure 10 Communication System State Machine Diagram

After the Communication System, the Information Recorder is meant to record the truth data that is generated by the UAV during a communication disconnect. For this project, the recorded information is not used, but it could be used to repair SA after reconnection. Following the Information Recorder is the System State Estimator (SSE). This estimates information elements when the UAV is disconnected from the OCS. It has three states, Standby, On, and Connected-On. It is in Standby when the UAV and OCS are connected, and the Target is within the field of view of the sensor. It is in the On state when the UAV and OCS are disconnected, and the Target could be within or outside of the WASS field of view. The Connected-On state represents when the Target is lost and outside of the WASS field of view, but the UAV and OCS are still connected. Estimation works by taking the last value of Burn Rate and Target Vector, and assuming they stay constant during the duration of the On or Connected-On states. A more sophisticated estimation technique could be used here by detecting trends and predicting based off those trends, but the simpler method works for this proof of concept.

After the SSE, the Divergence Detector takes those estimated values and compares them to the truth values generated from the Fuel System and WASS. When the divergence reaches a certain point, it sends signals to the Communication system that contain the divergence value as a parameter, where receive signal actions are waiting to receive them and pass that value into a property that is queued up for transmission after system reconnection. Refer to Figures 11 and 12 to see both sides of this transaction.

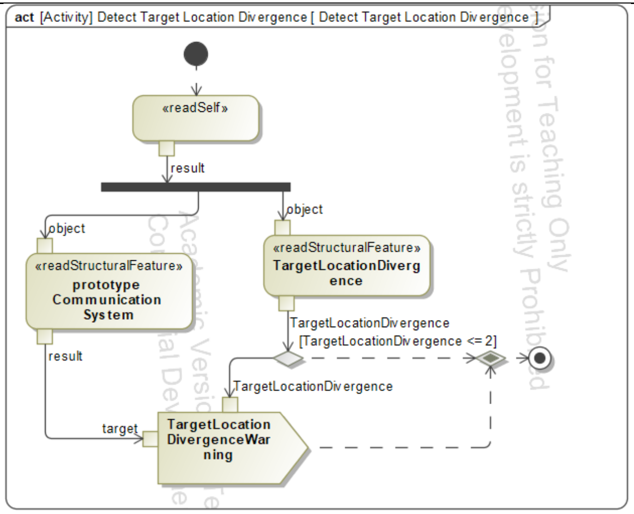

Figure 11 Detect Target Location Divergence

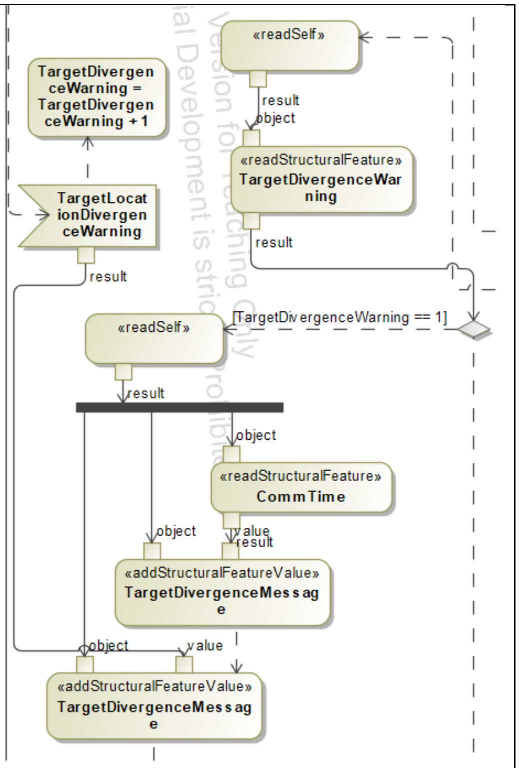

Figure 12 Receive Target Location Divergence

There are two pieces to the OCS - the OCS itself which displays information to the operator, and the OCS SSE, which calculates the system estimate during a disconnected or lost Target condition. The first behavior to execute is the OCS SSE, since that estimate needs to be generated before it can be displayed to the operator. It has two states, Standby and On, for when the OCS and UAV are connected and disconnected, respectively. It begins the On state at the same time as the UAV SSE, when the system experiences a communication disconnect. Then it uses the same estimate of the system state as the UAV SSE does. If the target is lost, but the system is still connected, the

OCS SSE remains in standby, since the OCS can receive the target location estimate from the UAV still.

When the OCS SSE is done executing, the next clock tick goes to the OCS. The primary function of the OCS is to display information to the operator, which is does during two states, “Display”, and “Display Estimated”. This was originally configured to receive signals with embedded parameters from the UAV Communication System, which in turn received those values from the relevant subsystems. The values from the signals were added into the “InformationDisplayed” value property, which has an undefined multiplicity, and collects all information during the simulation with a time stamp in front of it. Figure 14 shows how the Time value is added into “InformationDisplayed” right before the Fuel Quantity is added. Using signals to carry the information in parameters works to add the information into the “InformationDisplayed” value property, however there is a problem when exporting the information at the end of the simulation. The values of the parameters are not able to be exported into CSV files or Instance Tables the same way that numeric values are. Therefore, to have the information exported at the end of the simulation, read self and read property actions are used to directly read the relevant value properties (see Figure 15 for an example). The signals are still used to indicate whether the information is available to be read. They are used to increment value property counters, which are then used in guards to determine the execution path of the “Display Information Elements” activity, as seen in Figure 9. The state “Display Estimated” works in the same way as the first state, it just reads the estimated values instead of the truth values. That information is entered into the same

"InformationDisplayed" value property. Once the OCS behavior is executed, the flow goes back to the beginning of the "SysTick" activity in the Analysis Context, and the chain of behavior repeats until the Fuel Quantity goes to zero.

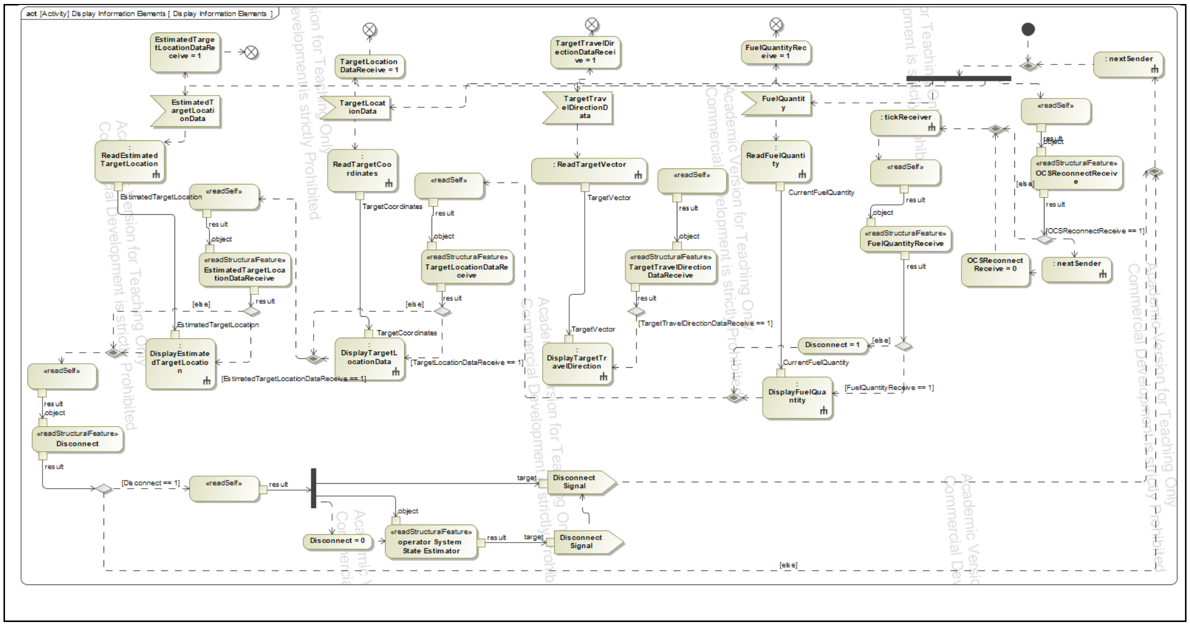

Figure 13 Display Information Elements Activity

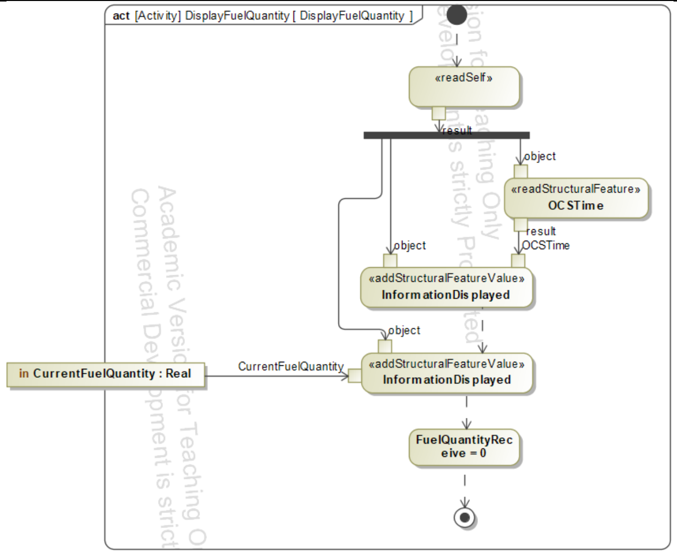

Figure 14 Display Fuel Quantity Activity

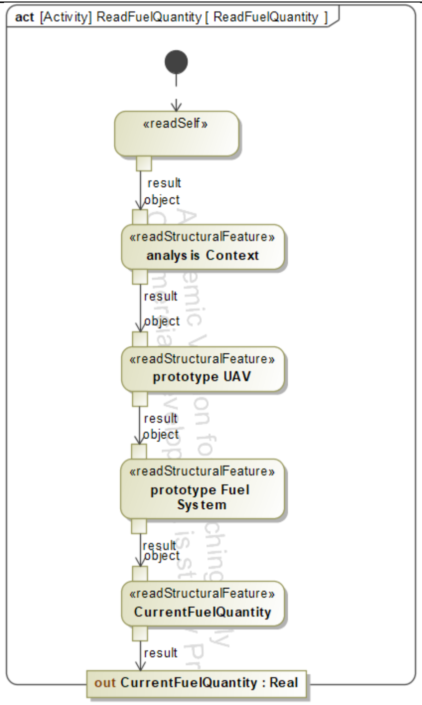

Figure 15 Read Fuel Quantity Activity

5. Results

The results of the simulation are exported into CSV files using the CSV Export capability built into Cameo. A CSV Export represents a block from the model, and any number of value properties of the block can be included. Those values can be read and written each time one changes during the simulation execution, or they can be read and written to the CSV file once at the completion of the simulation execution. If multiple values are listed in a CSV export, and the export is configured to write the value each time one changes, then each value property will be recorded every time one changes. This can be useful to note the "time" value each time the fuel quantity changes, for example, since the way the fuel quantity calculation works causes the value to change multiple times during each model "time" increment. Lastly to make it work, each CSV export is listed as an execution listener in the model's simulation configuration.

These exported results vary from run to run, since the target's movement and fuel burn rate have random functions built in. However, since this is just a proof of concept to test the idea of using Cameo and SysML to model this simulation environment, the actual values are not important. What is important is the demonstration of the values being generated in subsystems and passed to an operator, with the dynamic of disconnecting communications, system state estimations, and divergence detection all working together to provide more resilient SA than a traditional UAV system. Currently, this works by updating the operator with Fuel and Target Divergence information upon reconnection with the UAV, as well as at Target Lost warning should the sensor lose sight of the Target at any time during the simulation.

The following graphs show the results of a simulation run. Figure 16 shows the actual and estimated fuel quantities. The communication disconnect time was set at 50, and reconnect time at 100, so the estimation occurs between those times. The divergence threshold was set at 10%, and is represented by the red double arrows at time 71.

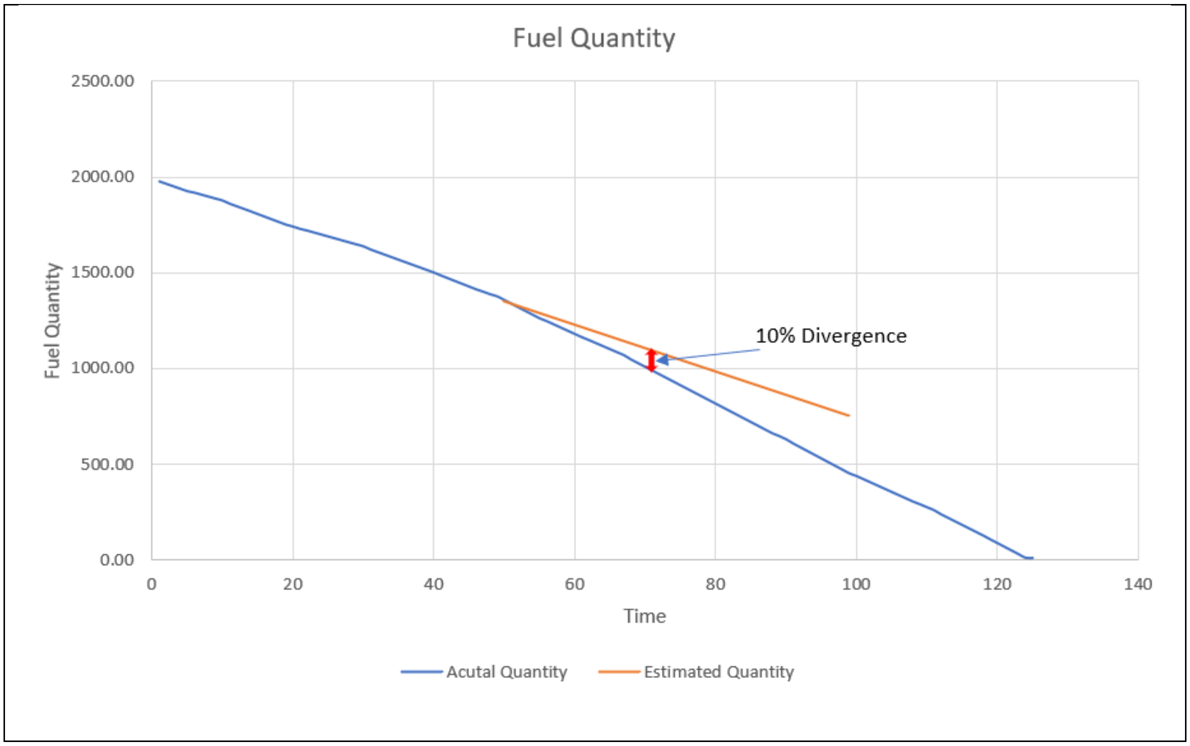

Figure 16 Fuel Quantity over Time

Figure 17 shows the information that is displayed to the operator during the simulation. The actual fuel quantity during disconnected communications is not shown since it was never displayed. The 10% divergence is shown to the operator upon reconnection and is shown in the graph as the red double arrows.

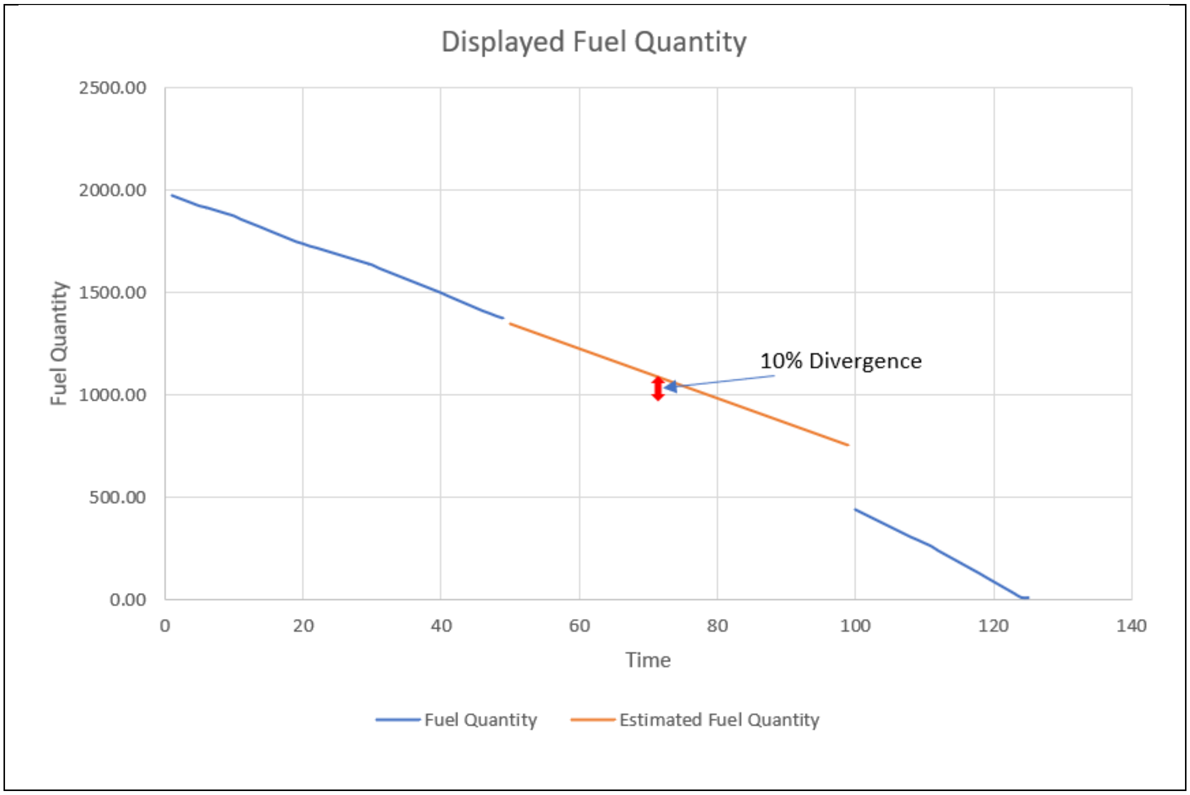

Figure 17 The Fuel Quantity that is displayed to the operator over time

Figure 18 shows the target's X and Y coordinates, proceeding from the origin. The target's location changed each time increment and each point on the graph represents a different location of the target. The estimated coordinates begin at (31.79, 30.78) when the communication loss occurs. The divergence threshold was set at 5 units for this simulation run, and that threshold is reached at the gray divergence point on the graph. The quarter-circle arc represents the field of view of the WASS and the yellow Target Lost Point represents the final known coordinate of the target, which occurred at time 82. The remainder of the blue points show the actual location of the target as it diverges from the target estimate. Since the communication reconnection occurs after the target leaves the field of view, the operator never sees the actual location again, and is only presented with the estimate.

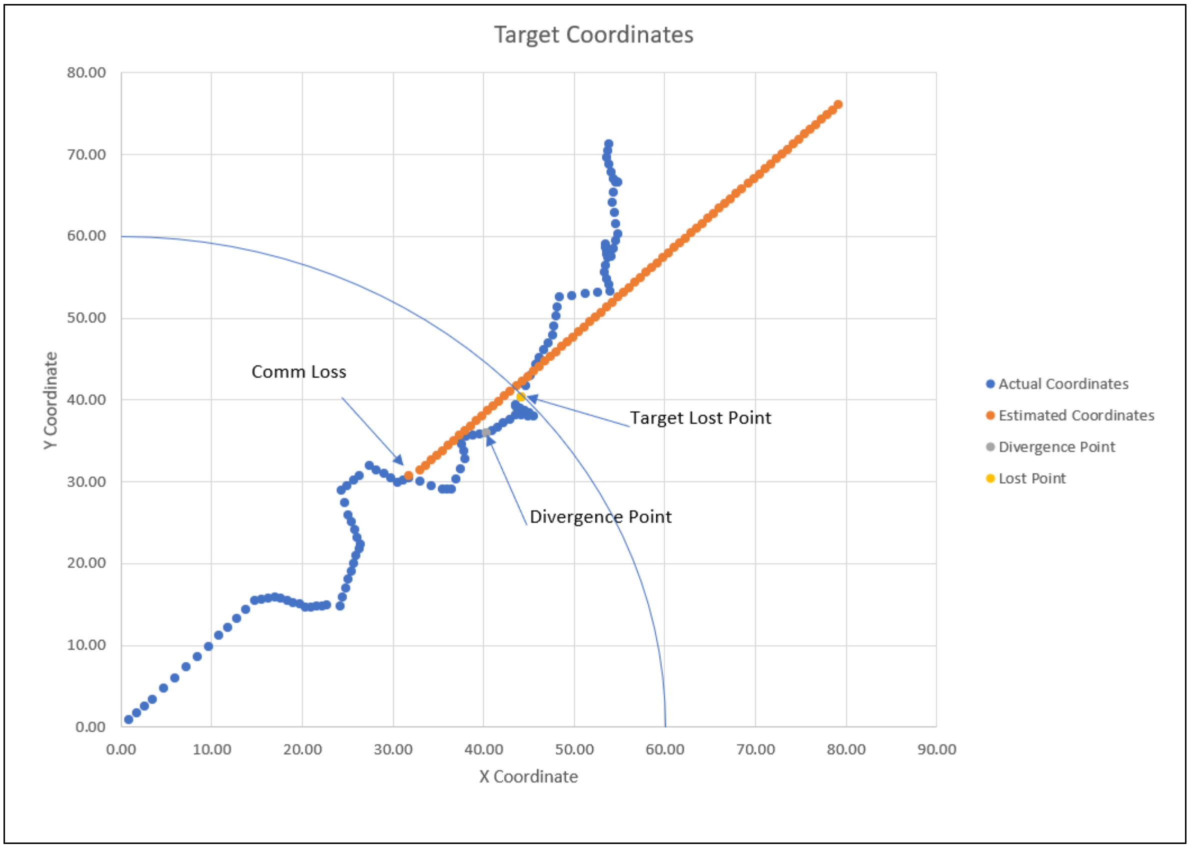

Figure 18 Target Coordinates

Figure 19 depicts only the information that is displayed to the operator relating to the target's location. When communication reconnection occurs, they are shown the estimated location of (63.59, 60.96), along with the point at which the UAV detected a divergence of 5 units and the last known location of the target at the Target Lost Point.

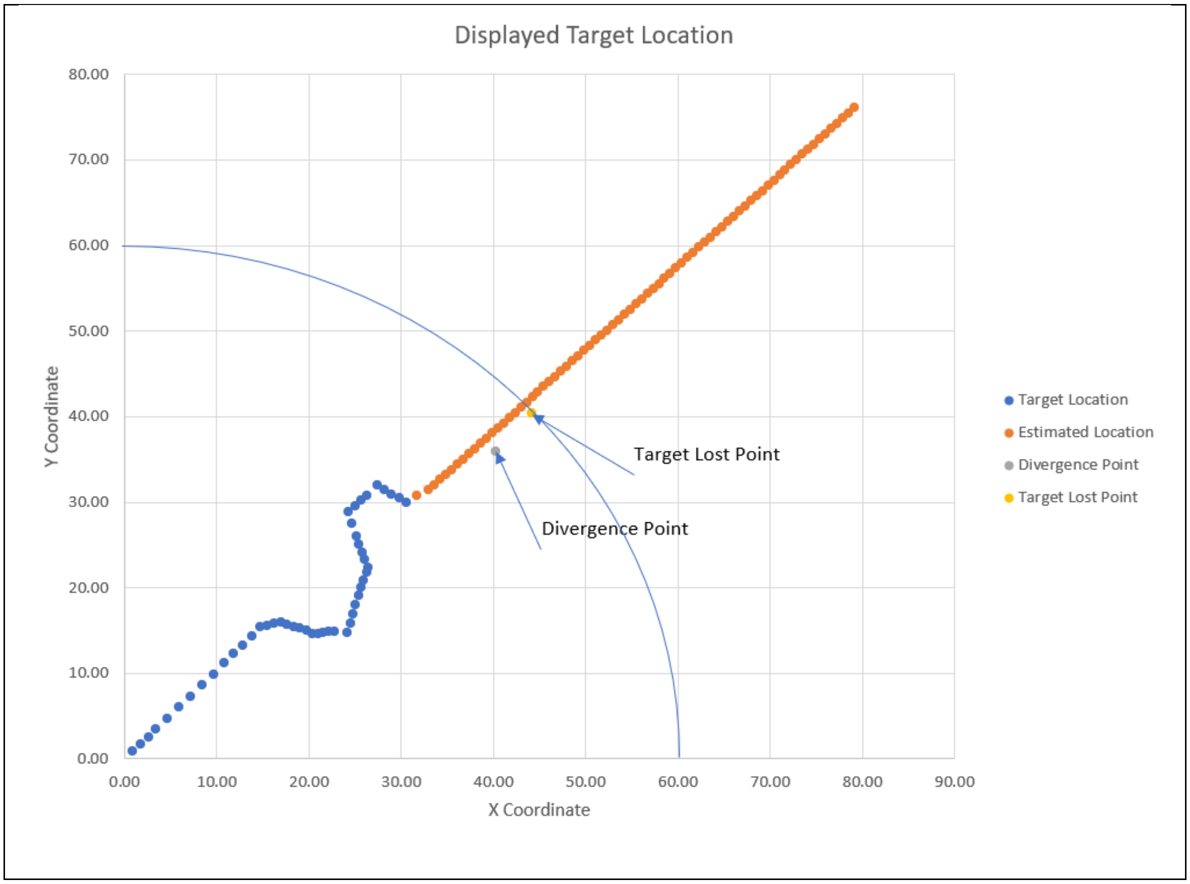

Figure 19 Target Location information that is displayed to the operator

Figures 20 and 21 show the information that would be displayed to the operator of a legacy system without estimation and SA repair capability. The only information displayed is the real-time data, and no warning messages like the Fuel Divergence, Target Divergence, or Target Lost messages, are relayed.

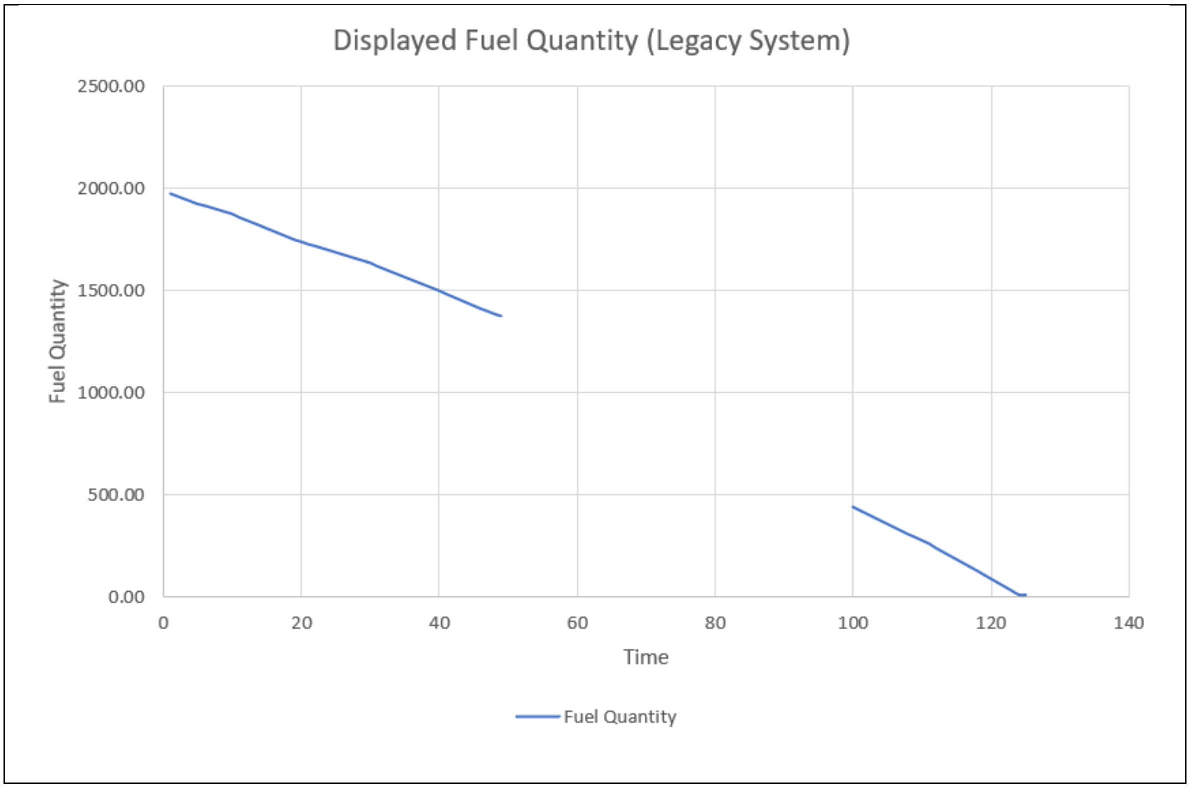

Figure 20 Fuel Quantity displayed to the operator of a legacy UAS

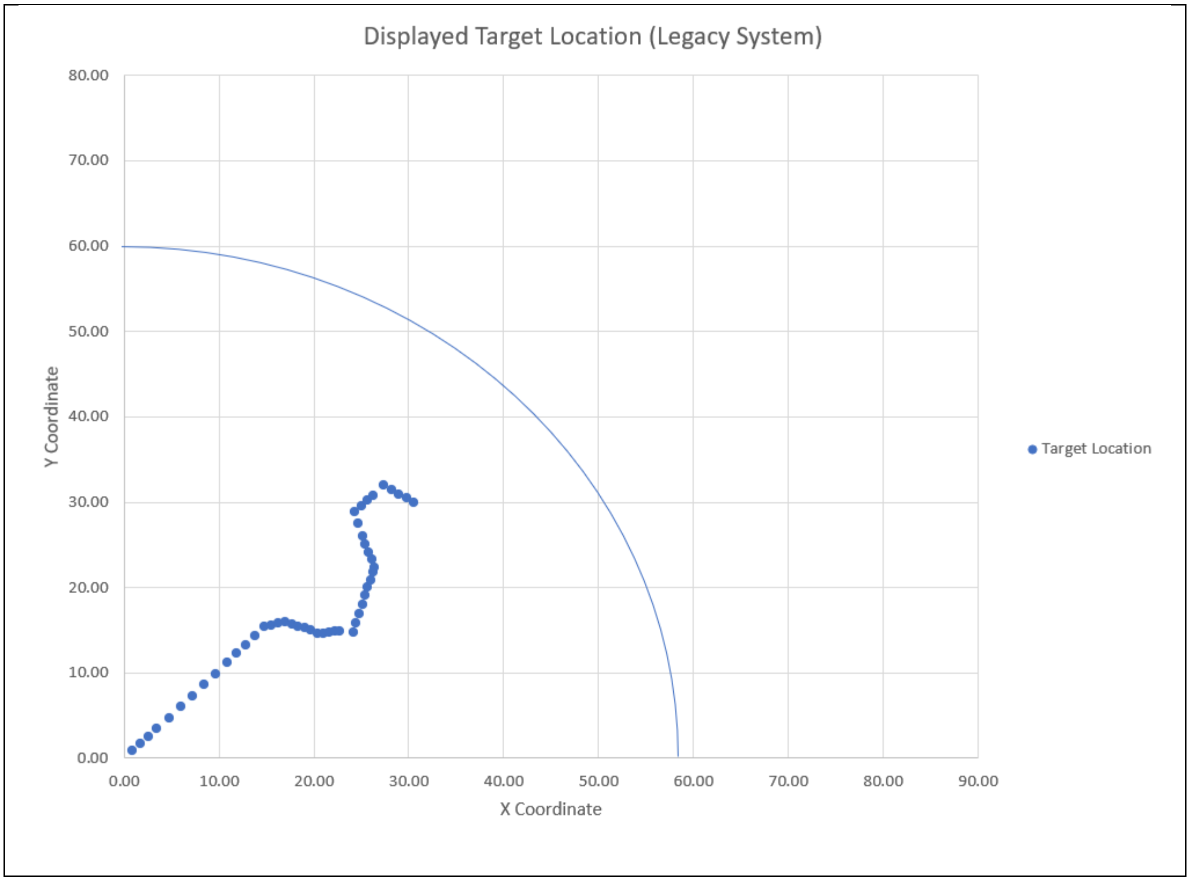

Figure 21 Target Location information that is displayed to the operator of a legacy UAS

7. Discussion

Many lessons can be taken away from using SysML and Cameo Systems Modeler to create a discrete event simulation environment. The ideal situation would be to have system blocks that can be easily interchanged with different environment blocks, depending on the scenario one wished examine. That way, a system can be modeled with certain capabilities that could then be put into different environments and conditions to see how it performs. This model accomplishes this goal to a certain extent but fails to achieve complete modularity. The way it is currently written allows some behavior to change on the system side or environment side while keeping the interfaces the same. For example, the target movement behavior could change so long as the result is still changing the same X and Y coordinate and vector value properties. Similarly, more sophisticated estimation logic could be implemented so long as the result is added to the estimated fuel quantity and estimated target location value properties. However, adding onto the system with different sub-systems or sensors which generate information elements, along with more environmental blocks, is still a labor-intensive process. This is because not only do the blocks need to be added with their own behavior, but the flow of information needs to be added through the Communication System to the OCS.

The other problem resolves around the use of the model-based clock. Care needs to be taken to ensure the behavior executes in the correct order, respecting all dependencies that each system may have with any other systems. As the overall system gets more complex and realistic, this could be an unwieldy and overly complex solution. Systems may need to execute different behavior in different places in the order of execution, and all of that needs to be carefully orchestrated by the modeler. Random erroneous behavior also surfaces with the signals being passed in the model. Occasionally, a signal will fail to be received at its destination, causing information to not be passed, or the entire simulation execution to stall. To mitigate that problem, multiple receive signal actions are placed in parallel leading to a merge, to increase the chances that a signal will be received. Further investigation is needed to understand why this happens, and what the risk of missing a signal is with any given number of receive signal actions. There is a chance that even with four or more receive signal actions, if the simulation is run over a long enough time, and if there are enough signals being sent, it may be hard to avoid the bug surfacing. Since this model is relatively simple and does not run for very long, placing extra receive signal actions proved to be enough to avoid the problem.

As mentioned before, there are multiple ways to get values around the system. Either they can be sent via parameters embedded in signals, or they can be read through read structural feature actions. In addition to the already mentioned problem of signal parameters not exporting in CSV files or Instance Tables, they also require great care to ensure the matching receive signal action is activated at the destination. This can be tricky when systems are also transitioning states at different times. This highlights the inherent disconnect between using signals, which behave asynchronously, with a model-based clock structure that relies on a careful order of events. Using read structural feature actions gets around both problems, although care must still be taken to ensure the value property being read has been updated during the same time increment, to ensure old data is not being read. This highlights the point made earlier about ensuring the correct behavior execution order.

References

[1] Dasgupta, S. (2017). Model-Based Clock.

https://docs.nomagic.com/display/CST185/Model-based+clock

[2] Smith, K. (2018). signals are being removed before being consumed.

https://community.nomagic.com/post14155.html

IV. Conclusions and Recommendations

Chapter Overview

This Chapter answers the research questions posed in Chapter 1. It also gives conclusions of this thesis based on the work and analysis done in the preceding conference papers, along with recommendations for future research.

Evaluation of Research Questions

1. *How can SysML and Cameo Systems Modeler be used to simulate a scenario where information is passed from the environment, through a UAV, and to an operator?*

a. *What mechanisms should be used to keep the system behavior synchronized with the environmental behavior?*

Any mechanism to keep the system behavior synchronized with the environmental behavior needs to be able to precisely control the order of events, not just between the system and environment, but within the sub-systems as well. This is because the desired result is time-stamped information being displayed to the operator, so the information that originates in the environment needs to be carried through the system within a certain time increment. It needs to be time-stamped so that the true, estimated, and displayed information can all be compared to one another. If there was no way to control this, one could never be certain that the information displayed to the operator represented the actual events that occurred at the specified time. The mechanism chosen in this thesis is the model-based clock. It allows fine-grained control over the order in which system and environmental behavior executes during a simulation. It does this by using receive and send signal actions at the beginning and end of each system's behavior. The receive signal action is what makes each system “wait its turn”, and the send signal action is what allows the simulation to move on to the next system behavior.

b. *How should information elements be stored and passed throughout different sub-systems of the model?*

Information elements are represented in the model by value properties of the blocks they belong to. For example, the X and Y coordinate of a target are value properties of that target block. These value properties can have multiplicities greater than one if appropriate, for example a target’s coordinates would have the X and Y coordinate as part of a single value property with a multiplicity of 2. These value properties can be passed from the environment and through different systems in two different ways, both of which are used and explored in this thesis. One way is to pass them as argument properties in signals, and the other is to use “read self” and “read structural feature” actions to read them from their sources to wherever they are needed. Due to the use of the model-based clock, either can be used. If the simulation were not synchronized with the clock, the signals would have a large advantage due to their asynchronous behavior. With the clock, the read actions occur in a specific order and are certain to read the correct value at the correct time.

Information can also be recorded using value properties with unspecified multiplicities. This is done in this thesis for the “information recorded” property that is a part of the “Information Recorder” system, as well as for the “information displayed” value property that is a part of the “Operator Control Station” system. In the case of the “information displayed”, the idea was to write out that value property at the end to an instance table where all the values could be read for analysis. A better method for recording results was discovered during the course of the thesis however. That is to export results of the simulation to comma separated value (CSV) files. Cameo allows values to be read and written to the file as they change during the simulation. Therefore, a large multiplicity value property does not have to exist to be read and exported at the end of the simulation.

c. How can information elements be estimated after communication loss between the UAV and operator control station (OCS)

This thesis uses simple estimation techniques for the relevant information elements, fuel quantity and target location. In the case of fuel quantity, it uses the last known burn rate and assumes it to be constant for the duration of communication loss. For target location, it uses the last known location and draws a line from the starting point (the origin in this case) to that location, then divides by the time. This averages the target's entire journey up until that point. These simple methods were used just to prove the concept. More sophisticated methods could also be used. The necessary information flow and capture is present in the model to allow for that. An area to explore further estimation is how to reference recorded data to calculate trends. This may be possible in Matlab or other external tools and would enable more accurate estimation that would improve SA by reducing the divergence from truth during communication loss.

2. In what ways can the results of the simulation be useful for analysis to improve the SA repair capability of the system design?

There are several ways that analysis of the simulation can help improve the SA repair design. One was already mentioned - improving estimation. Improvements in estimation will make the estimated values closer to the truth values, which will result in a less jarring transition after communications are re-established with the operator. Divergence warnings may not even be required if the estimation is good enough. Since estimation is a mostly mathematical process, incorporating those algorithms and analyzing improvements is well-suited for this type of model.

Another way that analysis can aid improved system design is by informing what other information elements may be useful to repair SA. For example, adding sensors to the UAV model to collect other information elements to help build better estimates, like using aircraft radar tracks to estimate winds (Hollister, Bradford, & Welch, 1986). This would be useful for similar reasons as increasing estimation fidelity, except by adding more data to the estimate, vs just having smarter estimation techniques.

Recommendations for Future Research

This thesis is as a test case for modeling and simulating an SA repair system, and there is still room for further research. First, a more complete UAV system could be modeled with a full set of information elements. The process of modeling the target location was appreciably different than modeling the fuel quantity, so it stands to reason that modeling other information elements could elucidate more lessons on how to model this type of system. It also introduces another element that needs to be considered – possible operator cognitive overload. More information elements would require that the system prioritize what is important to the operator to regain SA, because cognitive overload could reduce SA instead. To decide what is important to the operator, the system needs to know something about operator intent.

To inform operator intent, the model would need to include the dynamic of command and control over the UAV, which is currently missing and is another good area for further research. This UAV of the future is presumed to have advanced levels of automation that would be controlled at a higher level through something like Miller and Parasuraman's "playbook" method of supervisory control (2007). If the UAV in the model knew what the current and planned plays were, it could better prioritize the information to present to the operator during a loss of communication scenario. Since the UAV knows more about operator intent, it can draw the operator's attention to where the system has diverged from that intent, while filtering out irrelevant information. As an example, consider the fuel quantity information element. Depending on what the mission is and what plays have been sent to the UAV, fuel state divergence may or may not be an issue that the operator cares about. Filtering out that information would enhance the SA repair capability of the system. Incorporating operator commands in the model would be a substantial amount of work but would add a whole new layer that enables more useful design improvements.

Like the addition of operator commands, another layer that could be added is more environmental behavior that follows an operationally relevant scenario. Being able to model this is important because SA can best be understood in the context of specific scenarios. SAGAT measurements are taken during representative scenarios for this reason (Endsley, 2019). In particular, it is important to find ways to script specific actions taken by agents int he environment, both friendly and enemy. For simplicity, the current model uses random functions to change wind speed and target movement, but for specific scenarios that behavior would need to be repeatable for each run of the simulation.

Another area of further research is to find different ways to synchronize the behavior of every system in the model. The model-based clock is cumbersome to use, so any other method could make this method of modeling much more viable. One way may be to only synchronize the systems that generate information elements, while using signals to pass information to the other parts all the way down to the operator control station.

Additionally, more effort can be done to create SysML profiles to aid in the creation of SA repair system models and simulations. One such profile is a Unified Modeling Language (UML) profile called Modeling and Analysis of Real-Time and Embedded systems (MARTE). One of the domains suited for this profile is the reactive domain, where "Systems are generally tagged as "reactive" to stress the fact that they are meant to react to information inputs coming from some environment" (OMG, 2009, p. 8). This fits well with information elements coming from the environment as in this thesis. One of the subpackages of MARTE is Time Modeling, which "allows modeling of time and related structures, including concepts related to logical time, physical time, representation of instants representing time bases and occurrences of events over time" (Ribeiro, Pereira, Rettberg, & Soares, 2018). In a similar vein, there has been work in SysML-to-Arena transformation tools, where a system can be modeled in SysML using a domain specific language (DSL), then translated to an Arena™ model where it can be simulated in a discrete event simulation environment (Zülch, Gert & Börkircher, 2012). This method uses domain specific profiles to create the SysML model, along with stereotypes from a profile called SysML4Arena then uses model transformation tools to auto-translate stereotypes from SysML to Arena™ objects and processes (Zülch, Gert & Börkircher, 2012). The domain specific profile mentioned in this method could be an SA profile, which leads to another suggestion for future research. The creation of a SysML SA profile with domain-specific stereotypes would aid in the portability to other operational domains besides UAV operations. The ability to translate SysML models to a discrete event simulation program like Arena™ allows a system designer to use the best aspects of both modeling environments and could alleviate some of the difficulties and limitations of Cameo Systems Modeler discussed in this thesis.

Finally, it is worth stepping back to consider the big picture of what else could be required for an SA repair system. What else should be modeled to attain measures of Levels 1, 2, and 3 SA? What system blocks should be added to make the model of SA more complete? First, the System State Estimator could be re-made into an always-on system that continuously produces information elements to display based on whatever real-time information it is receiving. A continuous stream of accurate sensed information would increase the accuracy of the information displayed to the operator. When that stream of sensed information is interrupted, the SSE would continue to display an estimate based on the last known information. The other large piece that should be added is an information prioritization block, to prioritize information updates to the SSE when communication reconnection occurs. The prioritization should be based on bandwidth restrictions as well as relevance to the most critical information necessary for the operator to aid in the execution of the mission. Therefore, as discussed before, adding the operator and mission to the model is necessary to enable this prioritization.

Significance of Research

This research demonstrated some promise in using SysML for modeling SA repair in UAV operations, which ultimately is hindered by difficult execution using the available tool, Cameo Systems Modeler. The behavior must be synchronized with the model-based clock, which sometimes fails due to erratic signal behavior. The use of the clock makes the model relatively complex for a simple system. Using it for a more operationally realistic system and scenario would make it even more complex when the number of information elements and environmental tasks are considered. This calls into question the prospects of including a full system model into a digital thread. However, something like this could be used for simple systems where only a handful of information elements are of interest. Combined with some of the suggested future research, such a model could be useful for testing different SA repair system designs.

Bibliography

Endsley, M. R. (2019). A Systematic Review and Meta-Analysis of Direct Objective Measures of Situation Awareness: A Comparison of SAGAT and SPAM. Human Factors. https://doi.org/10.1177/0018720819875376

OMG. (2009). UML Profile for MARTE : Modeling and Analysis of Real-Time Embedded Systems v.1.0. Engineering, 15(November), 738. http://www.omg.org/spec/MARTE/1.1/

Ribeiro, F. G. C., Pereira, C. E., Rettberg, A., & Soares, M. S. (2018). Model-based requirements specification of real-time systems with UML, SysML and MARTE. Software and Systems Modeling, 17(1), 343–361. https://doi.org/10.1007/s10270-016-0525-1

Roper, W. (2020). Take the Red Pill: The New Digital Acquisition Reality.

United States Department of Defense. (2018). Summary of the 2018 Department of Defense Artificial Intelligence Strategy. 1–17. https://media.defense.gov/2019/Feb/12/2002088963/-1/-1/1/SUMMARY-OF-DOD-AI-STRATEGY.PDF

Uttamang, K. (2012). Simple Clock Modeling with SysML. SysML Users in Thailand. https://thaisysml.blogspot.com/2012/08/simple-clock-modeling-with-sysml.html

Zülch, Gert & Börkircher, M. (2012). Proceedings of the 2012 Winter Simulation Conference C. Laroque, J. Himmelspach, R. Pasupathy, O. Rose, and A.M. Uhrmacher, eds. Flexible Work Organization in Manufacturing – a Simulation-Supported Feasibility Study –, Meadows 1989, 1–12. http://ieeexplore.ieee.org.etechconricyt.idm.oclc.org/xpl/articleDetails.jsp?arnumber=6465229&tag=1

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

1. REPORT DATE *(DD-MM-YYYY)* 03-26-2021	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From – To)* Aug 2029 – March 2021

TITLE AND SUBTITLE Modeling Situation Awareness in a UAV Scenario Using SysML	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Hernandez, Tommy, R, Capt, USAF	**5d. PROJECT NUMBER**
	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/EN) 2950 Hobson Way WPAFB OH 45433-7765	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT-ENV-MS-21-M-234
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Intentionally left blank	**10. SPONSOR/MONITOR'S ACRONYM(S)**
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
DISTRUBTION STATEMENT A. APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES
This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

The U.S. Department of Defense (DoD) is rapidly moving towards using increasingly automated systems. Unmanned Aerial Systems (UAS) will be used in contested environments where direct human control may be impossible for long stretches of time. This increase in automation brings with it challenges in maintaining resilient Situation Awareness (SA) for operators, who still need to set goals for the UAS to accomplish their missions. Simultaneously, the DoD is increasing the adoption of Model-Based Systems Engineering (MBSE) in the development of weapon systems. This thesis explores the use of SysML to model of an SA repair system for a UAS and simulates it with the use of a model-based clock.

15. SUBJECT TERMS
Model-Based Systems Engineering (MBSE), Unmanned Aerial Systems (UAS), Situation Awareness (SA), autonomous systems

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Dr. Thomas C. Ford, AFIT/ENV
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)* 703-909-1151
U	U	U	UU	71	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18