Simulate Effects of Cyberspace Electromagnetic Activities (CEMA) in Mission Command Systems

Nathan Vey
U.S. Army Combat Capabilities Development Command - Soldier Center (CCDC-SC)
Simulation & Training Technology Center (STTC)
Orlando, Florida
nathan.l.vey.civ@mail.mil

J. Allen Geddes
Dynamic Animation Systems
Orlando, Florida
ageddes@d-a-s.com

Lawrence Elliott
Dynamic Animation Systems
Orlando, Florida
lelliott@d-a-s.com

Paul Tucker
Dynamic Animation Systems
Orlando, Florida
ptucker@d-a-s.com

ABSTRACT

The United States Army (USA) is developing and experimenting with concepts and force structures to conduct multi-domain operations (MDOs). The successful integration of cyberspace electromagnetic activities (CEMA) is a key tenant of winning an MDO as they affect, and are affected by, all of the warfighting functions. To effectively train for these operations, the USA requires capabilities to simulate CEMA and their effects on mission command systems. Several enhancements to enable training for CEMA in MDOs were made to a current "cyber for others" prototype training tool, Cyber Operations Battlefield Web Service (COBWebS), that was developed by the Army's Simulation and Training Technology Center (STTC), part of the Combat Capabilities Development Command – Soldier Center (CCDC – SC). The enhancements were funded by the Army Modeling and Simulation Office (AMSO) to improve the fidelity of the electronic warfare (EW) attack models that can stimulate live mission command systems and to provide a means to generate CEMA effects on Fires-related mission command systems (e.g., Advanced Field Artillery Tactical Data System [AFATDS]). This paper discusses the technical approach, successes, and shortfalls of integrating COBWebS with the Naval Research Laboratory's Builder tool to provide advanced radio frequency propagation models to simulate EW effects and with existing Call for Fire and AFATDS cyber training tools that are being developed for the Army's One Semi-Automated Forces (OneSAF) program.

ABOUT THE AUTHORS

Nathan Vey is a Science and Technology Manager at the U.S. Army Combat Capabilities Development Command Soldier Center Simulation and Training Technology Center (CCDC-SC STTC). He is a former Marine with operational experience in training Signals Intelligence (SIGINT) collection and analysis operations. Nathan's military training consisted of Electronic Intelligence (ELINT), Electronic Warfare (EW), and Geospatial Intelligence (GEOINT). He holds a Bachelor of Science (B.S.) in Electrical Engineering from the Milwaukee School of Engineering.

Lawrence Elliott is a Principal Software Engineer at Dynamic Animation Systems, Inc. He has over 15 years of Simulation Experience stemming from traditional programs of record such as WARSIM and CTIA to R&D efforts such as voice recognition, natural language processing, artificial intelligence, and space related technologies. He currently serves as the principal technical software lead for Cyber Operations Battlefield Web Service (COBWebS). He received a Bachelor of Science in Computer Science degree from Florida State University and a Master of Engineering with Concentration in Management from University of Florida.

J. Allen Geddes is a Software Engineer at Dynamic Animation Systems, Inc. He has over 15 years of Systems, Network, and Software Engineering experience and holds the following certifications: CompTIA A+, CompTIA Network+, CompTIA Security+, Microsoft Certified Professional (MCP), Microsoft Certified Systems Administrator (MCSA), and Microsoft Certified Systems Engineer (MCSE). He has earned an Associate of Science (A.S.) degree in Computer Programming and Analysis, a Bachelor of Science (B.S.) degree in Management Information Systems, a Bachelor of Applied Science (B.A.S.) degree in Software Development, and is currently pursuing a Master of Science (M.S.) degree in Modeling and Simulation at the University of Central Florida. Mr. Geddes currently works on various projects sponsored by the U.S. Army Combat Capabilities Development Command - Soldier Center (CCDC-SC) Simulation and Training Technology Center (STTC).

Paul Tucker is a Junior Software Engineer at Dynamic Animation Systems, Inc. He has 2 years of software development experience with 1 of those years being in the modeling and simulation industry. He currently works on the Cyber Operations Battlefield Web Service (COBWebS) which is sponsored by the U.S. Army Combat Capabilities Development Command - Soldier Center (CCDC-SC) Simulation and Training Technology Center (STTC). He has obtained a Bachelor's of Science in Computer Science degree from the University of Central Florida.

Simulate Effects of Cyberspace Electromagnetic Activities (CEMA) in Mission Command Systems

Nathan Vey
U.S. Army Combat Capabilities Development Command - Soldier Center (CCDC-SC)
Simulation & Training Technology Center (STTC)
Orlando, Florida
nathan.l.vey.civ@mail.mil

J. Allen Geddes
Dynamic Animation Systems
Orlando, Florida
ageddes@d-a-s.com

Lawrence Elliott
Dynamic Animation Systems
Orlando, Florida
lelliott@d-a-s.com

Paul Tucker
Dynamic Animation Systems
Orlando, Florida
ptucker@d-a-s.com

INTRODUCTION

Since cyberspace has been recognized as a warfighting domain, the Army Modeling & Simulation Office (AMSO) has sought to provide the six modeling and simulation (M&S) enabled communities (acquisition, analysis, experimentation, intelligence, test and evaluation, and training) with the necessary M&S tools and capabilities to replicate cyberspace and electromagnetic activities (CEMA) for their needs. Our team proposed enhancements to an existing M&S CEMA capability called COBWebS, which stands for Cyber Operations Battlefield Web Service, to AMSO to help meet some of the desires of the communities.

Initially, the Army's Simulation and Training Technology Center (STTC), within the Combat Capabilities Development Command – Soldier Center (CCDC-SC), developed COBWebS as an Information Assurance (IA) compliant web-based software application capable of simulating effects of CEMA on command and control (C2) communications between simulated, synthetic entities and live Mission Command Information Systems (MCIS) on the tactical network. COBWebS provides a Cyber Role Player the ability to inject Information Interception (II), Information Delay (ID), Information Forgery (IF), and Denial of Service (DoS) attack effects on live MCIS that are part of an M&S-enabled exercise, creating the effects of CEMA attacks. Although a successful prototype, COBWebS still was not able to meet some of the communities' needs for affecting MCIS.

The proposed enhancements to COBWebS included improving the fidelity of electronic warfare (EW) attack modeling within COBWebS, and adding functionality for generating CEMA attack effects on the Army's Fires-related mission command systems (e.g., Advanced Field Artillery Tactical Data System [AFATDS]). Due to the current attention focused on the cyberspace domain and the need to provide realistic simulation environments in order to conduct Multi-Domain Operations (MDO), AMSO selected and sponsored this project. The remainder of this paper describes our approach to enhancing COBWebS with these new capabilities.

ELECTRONIC WARFARE (EW)

Currently, there are a few low-fidelity ways for incorporating EW attack effects in M&S-enabled exercises that use live mission command devices. This is typically accomplished by scripting EW attack scenarios using "white cards" that describe the EW attack scenario to the operator, but do not generate the effects on their mission command device. Sometimes, trainers physically manipulate network equipment, such as unplugging a cable from a network switch, to create the effects of total communication loss due to an EW jamming attack. Unplugging a network cable, however, creates an all-or-nothing effect on the mission command device, whereas a real-world EW jamming attack might have a more intermittent data loss effect.

To address this limitation, the initial version of COBWebS provided a rudimentary EW jamming attack effect capability that let the Cyber Role Player launch area-based DoS attacks by drawing an area on the COBWebS Cyber Editor map interface, and COBWebS uniformly denied a specified percentage of tactical messages originating within that area. For example, if the Cyber Role Player launched a circular DoS attack with a 1 km radius and an 80-percent denial value, COBWebS would uniformly deny 80-percent of the tactical messages originating anywhere within the 3.14 km^2 DoS attack area, and COBWebS would transmit 20-percent of tactical messages originating within the same area, through to the mission command devices on the tactical network.

This capability let Cyber Role Players specify the precise location and intensity of EW attacks on the mission command devices, but still did not represent real-world EW attack effects with very high fidelity.

Radio Frequency (RF) Propagation Modeling

To improve the fidelity of EW jamming attack effects within COBWebS, we needed to incorporate various environmental factors such as terrain and elevation, as well as antenna and frequency properties, into the calculations. These additional parameters would allow us to more accurately and intelligently determine whether or not EW jamming attacks would impact tactical messages. Instead of developing these complex radio frequency (RF) propagation models from scratch, we found that there are existing RF propagation modeling solutions that we could leverage, such as the Electromagnetic Propagation Integrated Resource Environment (EMPIRE).

EMPIRE is a suite of government off-the-shelf (GOTS) and commercial off-the-shelf (COTS) RF propagation models, used to predict electromagnetic field propagation over a wide range of conditions, environments, and frequencies. Numerous government Tactical Decision Aids (TDAs) use EMPIRE as their electromagnetic calculation engine. Figure 1 shows a high-level overview of the EMPIRE suite of propagation models, and how a TDA can utilize EMPIRE for RF propagation calculation and analysis. (Remcom, 2019)

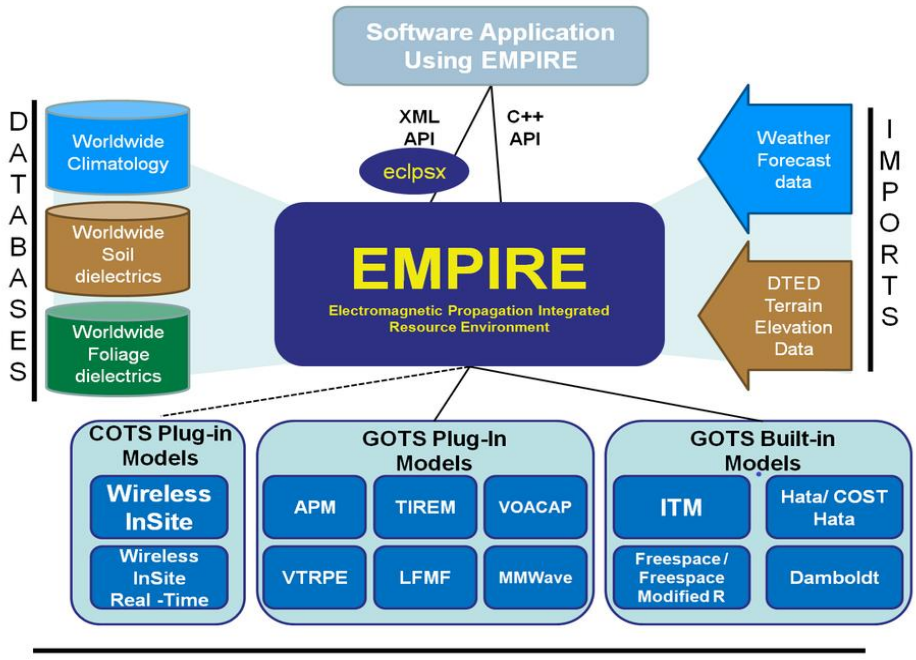

Figure 1. Propagation Models

Naval Research Laboratory Interactive Scenario Builder (NRL Builder)

One such TDA that utilizes the EMPIRE suite of RF propagation models is a GOTS product developed by the Naval Research Laboratory (NRL) called Interactive Scenario Builder, or NRL Builder. From the product description in the NRL Builder User Manual, NRL Builder is *"a computer simulation tool that provides a graphical depiction and*

detailed analysis of radio frequency (RF) propagation. This includes one and two-way communication links (e.g., communication radios, cellular phone towers, radio broadcasts), probability of detection by radar, communications and radar jamming capabilities of platforms in addition to providing geo-spatial and temporal situation awareness (SA). It incorporates complex antenna pattern data as well as the effects of meteorology, terrain, environment, and countermeasures when computing RF propagation values. It visualizes and is compatible with different map products including NGA (CADRG and CIB) and Google Earth (KML) thereby enhancing geo-spatial SA for a user. Builder can be used for pre-mission planning, real-time situational awareness, and after-action debriefing." (Naval Research Laboratory, 2018)

COBWebS - NRL Builder Interface

NRL Builder provides multiple different application programming interfaces (APIs) to be able to calculate and retrieve signal strength values and radio frequency propagation loss values from an external application. During our initial research into using NRL Builder, we were not sure which NRL Builder interface we should utilize to be able to improve the EW jamming attack models within COBWebS so we started with the Environment and Propagation Library (EPL) Application Programmable Interface (API) that NRL Builder provides.

The EPL API is built on top of an open source Remote Procedure Call (RPC) framework called gRPC, that allows NRL Builder procedures to be called from an external application that could be running locally or on a remote machine. The EPL API provides functionality for calculating one-way and two-way radio frequency propagation loss values when you provide transmitter and receiver properties as input parameters for the API call. Since our use-case would generate a large number of API calls at a rapid pace in order to evaluate whether or not simulated tactical messages should be impacted by EW jamming attacks, we had concerns if this API would work as we did not want to introduce a bottleneck or unacceptable latency in an exercise due to additional data calls to the EPL API.

We discussed these questions and concerns in detail with the NRL Builder support team. We described our high-level requirement to improve the EW modeling capabilities within COBWebS, and we described how we were attempting to use NRL Builder's EPL API to accomplish this. For our use-case, the NRL Builder support team recommended that we use the NRL Builder Web Service instead of the EPL API.

The NRL Builder Web Service provides the ability to generate signal strength plots. A signal strength plot contains a dataset of numerous signal strength values in a single API call response, which would drastically cut down on the number of API calls COBWebS would need to make to NRL Builder. Using this service, COBWebS would only make a single API call for an EW jamming attack, instead of having to make individual API calls for every tactical message it receives from the simulation; the next section contains additional detail on this. The web service was chosen for our use-case as it better met our needs for an M&S training system than what the EPL API could do. The NRL Builder Web Service that the NRL Builder support team referred us to download contained the web service in addition to a sample client application that can stimulate the NRL Builder Web Service.

Unfortunately, the NRL Builder website did not provide any documentation on how to utilize the NRL Builder Web Service. However, we analyzed the network traffic between the provided NRL Builder Web Service and sample client application and were able to use that information to construct a comprehensive list of all HTTP requests and responses that the NRL Builder Web Service could handle. With this information, we were able to successfully generate signal strength plots and process the response data from COBWebS.

Signal Strength Plots

When generating a signal strength plot, NRL Builder calculates what a transmitter's signal strength would be at numerous sample points (potentially thousands) relative to the transmitter's location. NRL Builder then returns a dataset containing all of the sample point signal strength values, along with a corresponding image file that contains a visualization of the signal strength plot.

When we submit an HTTP request to NRL Builder to generate a signal strength plot from COBWebS, we specify input parameters such as the latitude and longitude coordinates where the jamming transmitter is located, the height of the transmitter, the radius of the plot we would like to generate, and the number of radials and the number of points along each radial we would like NRL Builder to sample for the plot.

We also tell NRL Builder which RF propagation model we would like to use when polling the terrain and calculating the signal strength values at each sample point. For our use-case, the NRL Builder support team recommended that we use the EMPIRE Terrain Integrated Rough Earth Model (TIREM) RF propagation model, which factors in numerous environmental variables, antenna variables, and is accurate above land-based terrain.

Figure 2 shows an example signal strength plot with 36 radials and 200 sample points along each radial. In this example, NRL Builder polls the terrain with the provided input parameters and generates a dataset of 7,200 signal strength sample point values which the web service returns to COBWebS.

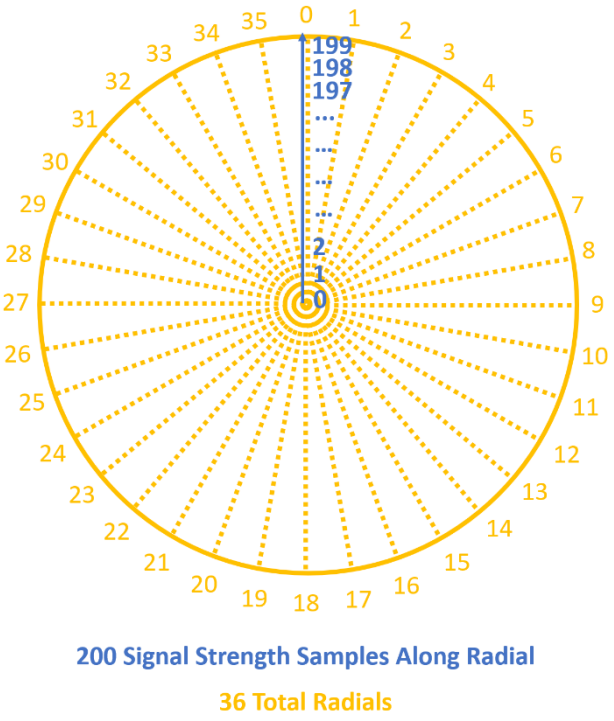

Figure 2. Signal Strength Plot Sample Point Example

By adjusting the number of radials and the number of points along each radial through the COBWebS interface, the Cyber Role Player can indirectly manipulate the fidelity and the processing speed of the EW jamming attacks.

While NRL Builder is generating the signal strength plot, COBWebS queries the NRL Builder Web Service once per second for plot processing status and displays the progress information to the user on the COBWebS Cyber Editor map interface.

When the signal strength plot processing completes, COBWebS makes another API call to the NRL Builder Web Service to retrieve the complete signal strength sample point dataset, which the web service returns in comma-separated value (CSV) format. COBWebS then makes an API call to the NRL Builder Web Service to retrieve the corresponding signal strength intensity imagery files, which are returned in Keyhole Markup language Zipped (KMZ) format. Once received, the contents of the KMZ zipped file are extracted and the EW attack intensity imagery is displayed on the COBWebS Cyber Editor map. The display provides the Cyber Role Player with a visual indication of the EW attack intensity at various locations on the map.

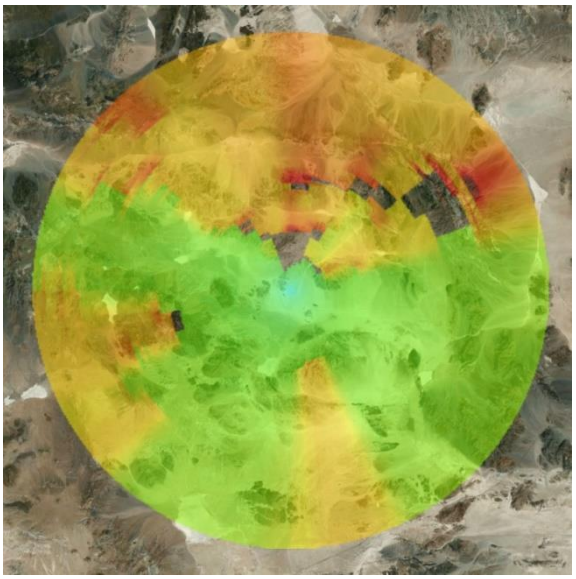

Figure 3. Signal Strength Plot Image

Figure 3 shows as an example of a signal strength plot intensity image returned by the NRL Builder Web Service.

Converting dBm to Denial of Service Percentage

Once COBWebS receives the EW attack signal strength values from the NRL Builder Web Service, the CSV file is parsed and the signal strength values, measured in decibel-milliwatts (dBm), are stored locally for rapid lookup. This ensures fast performance as COBWebS receives tactical messages from the simulation.

For each tactical message that COBWebS receives from the simulation, COBWebS needs to evaluate whether or not the tactical message is impacted by the EW attack, and if so, what the signal strength value of the EW attack is at the closest sample point that we have to that tactical message.

Figure 4 shows an example of how we would look up the closest EW attack signal strength sample point (■) value (dBm) to a tactical message (●), in three steps:

Step 1: Calculate the **bearing** between the tactical message (●) and the EW attack center in order to identify which radial the tactical message (●) is closest to.

Step 2: Calculate the **distance** between the tactical message (●) and the EW attack center to find the closest sample point along the radial.

Step 3: Look-up the **signal strength value (dBm)** of the EW attack at the closest sample point (■) to the tactical message (●).

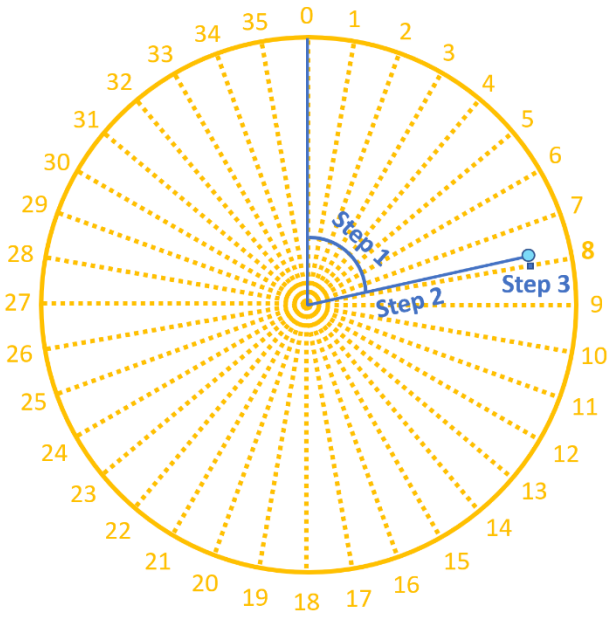

Figure 4. Signal Strength Value (dBm) Look-Up

Once we have the signal strength value (dBm) of the closest sample point, we plug that value, along with the minimum and maximum signal strength values into the formula in Figure 5 below, which is used for converting RF signal values (dBm) to signal strength percentages (Granados, 2016).

$$percent = 100 \times (1 - (P_{dBm\_max} - P_{dBm}) / (P_{dBm\_max} - P_{dBm\_min}))$$

Figure 5. Signal Strength Value (dBm) to Percentage Formula

Finally, once the corresponding signal strength percentage is known for the EW attack at the location of the tactical message, this percentage is used to evaluate whether the tactical message should be transmitted or denied. In other words, this signal strength percentage equates to the probability COBWebS will deny transmission of the tactical message through to the tactical network as a result of the EW attack.

For example, if the signal strength percentage is 100, there is a 100 percent chance that the EW attack impacted the tactical message and a 100 percent change that COBWebS will deny the message. If the signal strength percentage is 80, there is an 80 percent chance that the tactical message is impacted by the EW attack and an 80 percent chance that COBWebS will deny the message, and so on.

Summary of EW Enhancements to COBWebS

The initial version of COBWebS provided rudimentary EW attack modeling capabilities that let the Cyber Role Player specify the precise location and intensity of EW attacks on the mission command devices. This capability was an improvement to the current state of incorporating EW attacks into exercises, but did not necessarily represent real-world EW attack effects with very high fidelity.

With the new NRL Builder interface, COBWebS can now vary the percentage that it will deny tactical messages coming from the simulation through to the tactical network in a more intelligent and realistic manner that incorporates environmental factors such as terrain and elevation, as well as antenna and frequency characteristics.

This new capability was tested using One Semi-Automated Forces (OneSAF) to stimulate a full suite of U.S. Army mission command systems. The test verified the presence of EW jamming attack effects on the Army mission command systems. It was confirmed that the entities further from the EW attacks or entities with obstructions between themselves and the EW attack transmitter had less impact from the jamming attack. Also, as expected, entities that were closer to the origin of the EW attack or with direct line-of-sight to the EW jamming transmitter had a stronger impact from the attack, and the end result is more realistic EW attack effects on the mission command systems.

Potential future work could take it a step further and let the user define the EW attack jammer properties (instead of using the emitters available and pre-defined in the NRL Builder database). The calculations could also potentially take simulated entity transmitter and receiver equipment properties and frequencies into consideration when evaluating whether or not a tactical message should be impacted by an EW jamming attack.

CEMA ON FIELD ARTILLERY FIRE MISSIONS

The second objective of this project was to introduce CEMA attack effects into the Army's field artillery fire mission messaging chain. When conducting artillery fire missions, the various entities involved in initiating and executing the mission exchange a series of messages. Typically, a forward observer (FO) initiates a "call for fire" (CFF) request for artillery fires on a specific target location. The Fire Direction Center (FDC) receives the CFF message and determines what ammunition should be used, what guns are available to execute the mission, and then the FDC sends the mission to the individual guns to fire. The FO then observes the initial volley and provides adjustment instructions as necessary to the FDC until the guns are firing precisely on the target. (Headquarters, Department of the Army, 1991)

To introduce CEMA attack effects on field artillery fire missions, it was necessary to find a way to intercept and modify these messages as they are exchanged between the various mission command systems involved in the messaging chain. The mission command system that the Army uses for planning, coordinating, controlling, and executing fires missions is the Advanced Field Artillery Tactical Data System (AFATDS). The communication protocol between live AFATDS devices is proprietary and encrypted so we were unable to introduce CEMA attacks on fire mission messaging between multiple live AFATDS devices. We were, however, able to intercept and modify these messages as they pass from a simulated AFATDS device to a live AFATDS device. And we were able to accomplish this in two different ways using two different interfaces and tactical message flows, which the following section describes in further detail.

Tactical Message Manipulation using the Mission Command Adapter Interface

The Mission Command Adapter (MCA) is a web service used by the Army for translating messages coming from synthetic entities in a simulation through to live mission command devices on the tactical network. Using the MCA, the Army can stimulate live mission command devices using a constructive simulation rather than needing actual live entities. With its current architecture COBWebS can intercept, delay, deny, and forge data between the simulation and the MCA to create CEMA attack effects on the mission command devices on the tactical network. The initial version of COBWebS only allowed for forgery and manipulation of tactical messages such as Entity Position Reports, Observation Reports, and Free Text, but for this project we updated COBWebS to be able to forge and manipulate CFF tactical messages as well.

To accomplish this, we created a "tactical message queue" that lets COBWebS intercept tactical messages coming from the simulation, and queue them instead of dispatching them directly through to the MCA. Using the updated COBWebS Cyber Editor user interface, the Cyber Role Player can review and modify the properties of queued messages before releasing the messages through to the MCA. The initial implementation only supports the queueing of CFF tactical messages, but the capability can support queueing any type of tactical messages as needed for future use-cases and requirements due to its flexible architecture.

The CFF MCA tactical message queue can be in one of three different states: **Off**, **Global mode**, or **Information Interception (II) mode**. When the CFF MCA tactical message queue is **Off** COBWebS does not queue any CFF

tactical messages and automatically dispatches all CFF tactical messages through to the MCA. When the CFF MCA tactical message queue is in **Global mode**, COBWebS intercepts and queues all CFF tactical messages until the user manually releases them through to the MCA. When the CFF tactical message queue is in **II mode**, COBWebS intercepts and queues any CFF tactical messages that are under an active II attack. (See Figure 6)

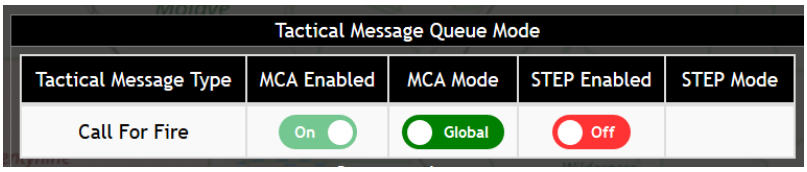

Figure 6. Call for Fire Tactical Message Queue Controls

Putting the tactical message queue in II mode lets the Cyber Role Player focus the "cyber playbox" within the exercise to a specific geographic area or a specific set of entities, so that they are not disrupting all CFF tactical messages in the entire exercise, which could have unintended consequences and detrimental effects in a large exercise.

When the CFF MCA tactical message queue is enabled and the user initiates a CFF mission from within the simulation that is destined for a live AFATDS device, the following occurs:

Step 1: User initiates a CFF mission from within the simulation. Note the original target coordinates in Figure 7.

Figure 7. Simulation Call for Fire Mission

Step 2: COBWebS intercepts and queues the original CFF tactical message (See Figure 8).

Figure 8. Call for Fire Tactical Message Queue

Step 3: The Cyber Role Player changes properties of the original CFF tactical message. In this example the user has modified the target location coordinates by clicking the push-pin icon and then selecting a new location on the map. Note that the Cyber Role Player can change any of the Basic or Advanced properties seen in Figure 9.

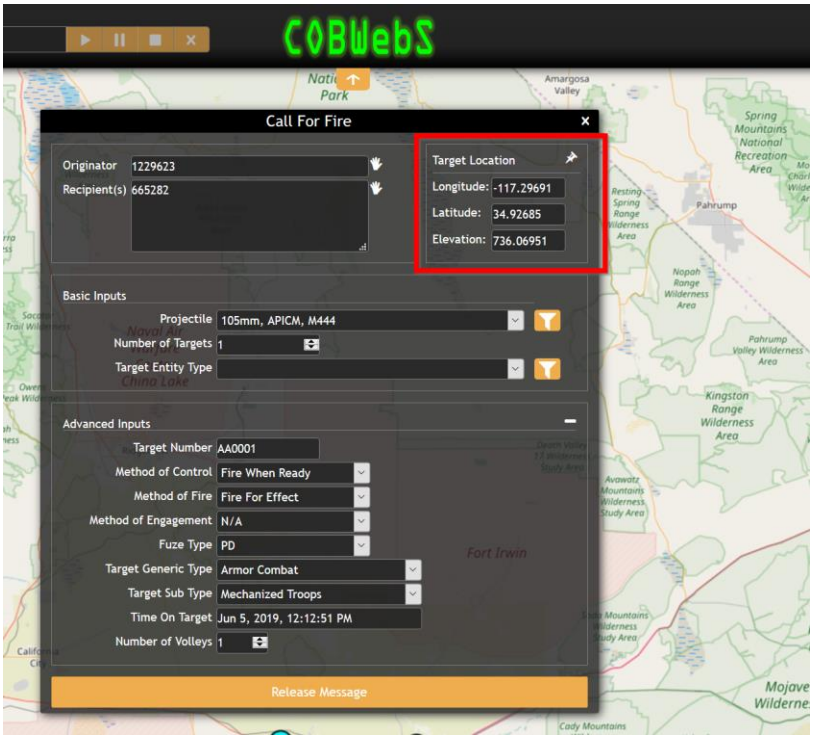

Figure 9. Call for Fire Tactical Message Manipulation

Step 4: The Cyber Role Player then releases the modified CFF tactical message through to the MCA, and the modified CFF tactical message properties are visible in the MCA message flow (See Figure 10).

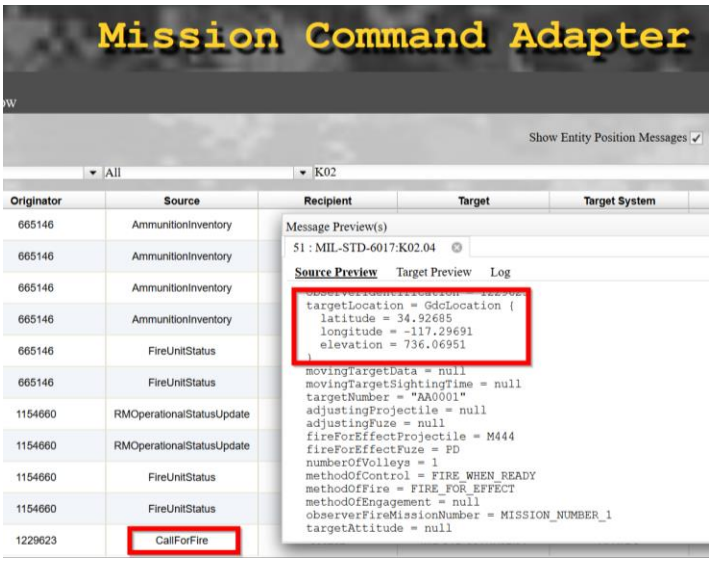

Figure 10. Modified Call for Fire message in MCA Message Flow

As you can see in the above example, the Cyber Role Player intercepted the queued CFF message, modified the target coordinates, and then released the modified CFF message through to the Mission Command Adapter which sends it out to live AFATDS devices on the tactical network with the modified target coordinates.

Tactical Message Manipulation using the Carnegie Mellon University STEP Interface

In addition to the above functionality, the effort enhanced COBWebS to be able to modify and manipulate CFF tactical messages exchanged between the simulation and Carnegie Mellon University's (CMU) Simulation, Training, and Exercise Platform (STEP) live cyber range. This portion of the project leveraged a previous effort to interface the OneSAF constructive simulation with the STEP live cyber range using CMU's Cyber Kinetic Environment Interface (CKI) which was also an AMSO funded project. The goal for that effort was to enable the exchange of cyber-related event data between the STEP live cyber range and the OneSAF simulation, so that cyber-events occurring on the live cyber range environment could impact models and behaviors inside the simulation, and vice versa.

In order to interact with the simulation over this same STEP interface, a RESTful STEP interface to COBWebS was added that allows COBWebS to participate as a federate in exercises that are using the STEP interface. Once that interface to STEP working, functionality was added to COBWebS to intercept and queue CFF tactical messages coming over the STEP interface, in the same way that we intercepted and queued CFF tactical messages coming over the MCA web service interface, discussed in the previous section. The difference here is that when COBWebS releases a queued STEP CFF tactical message, it now goes back into the simulation over the STEP interface instead of out to the tactical network through the MCA. This lets the user see the effects of the CEMA attack on the CFF tactical messages inside the simulation, as opposed to in the MCA message flow or on the live devices on the tactical network.

The CFF STEP tactical message queue can also be in one of three different states: **Off**, **Global mode**, or **Information Interception (II) mode**. And these modes behave the same way as the CFF MCA tactical message queue, described in the previous section (See Figure 11).

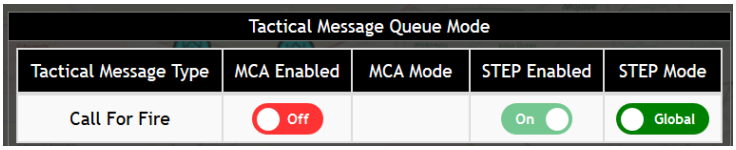

Figure 11. Call for Fire STEP Tactical Message Queue Mode

When the CFF STEP tactical message queue is enabled and the user initiates a CFF mission from within the simulation that is destined for an external AFATDS device on the STEP live cyber range, the following occurs:

Step 1: User initiates a CFF mission from within the simulation. Note the original target coordinates in Figure 12.

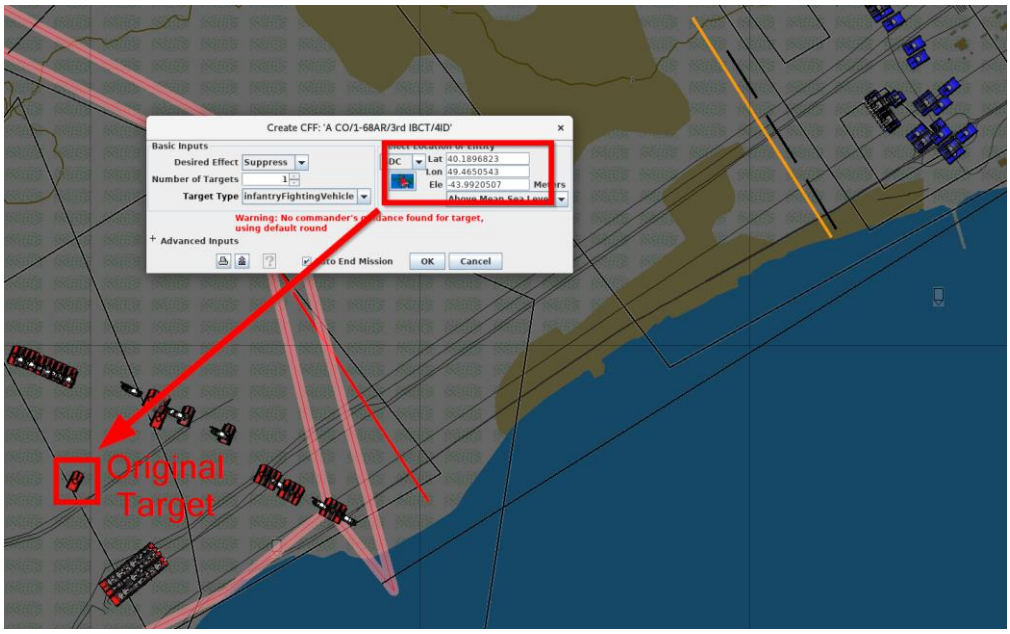

Figure 12. Simulation Call for Fire Mission

Step 2: COBWebS intercepts and queues the original CFF tactical message (See Figure 13).

Figure 13. Call for Fire Tactical Message Queue

Step 3: The Cyber Role Player changes properties of the original CFF tactical message. In this example, the user modified the target coordinates by clicking the push-pin icon and then selecting a new location on the map (See Figure 14).

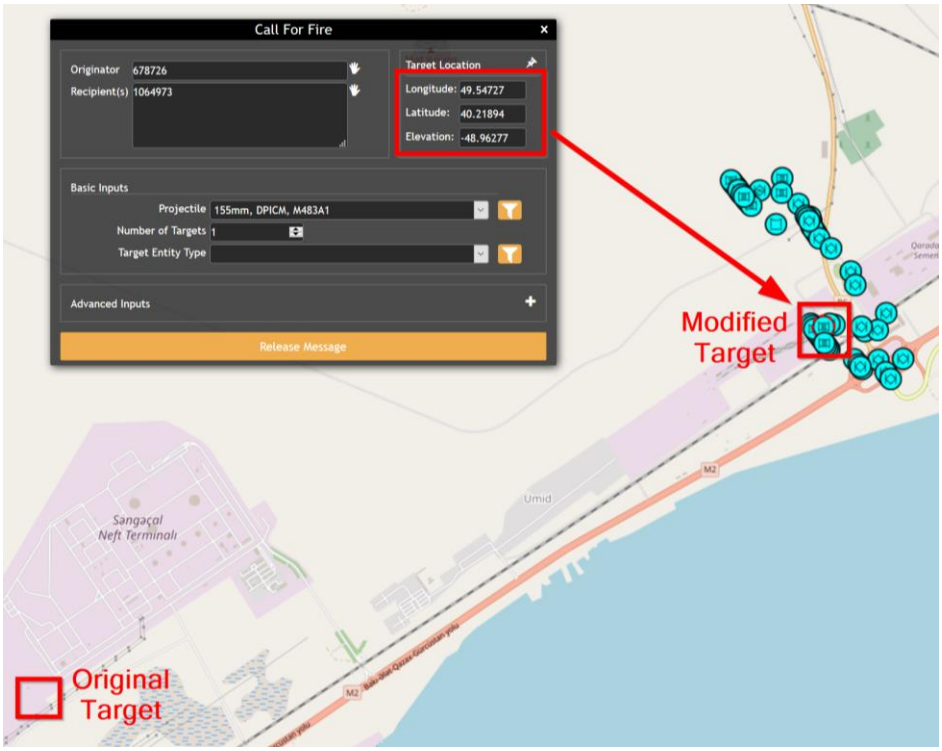

Figure 14. Call for Fire Tactical Message Manipulation

Step 4: The Cyber Role Player releases the modified CFF tactical message back to the simulation over the STEP interface, where the fire mission completes by firing on the modified target (See Figure 15).

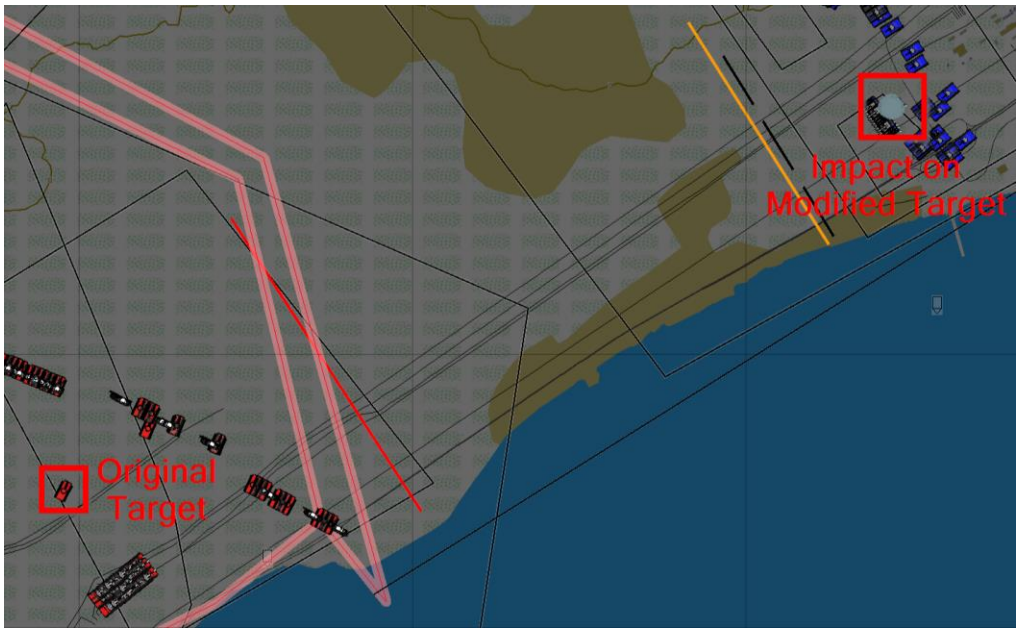

Figure 15. Call for Fire Tactical Message Manipulation Effects in Simulation

Summary of COBWebS Effects for Field Artillery Fire Missions

COBWebS has been enhanced to be able to incorporate CEMA attack effects on fire missions using both the MCA interface and the STEP interface in order to better close the gap identified by AMSO. The enhanced capability allows Cyber Role Players to incorporate CEMA attack effects on live AFATDS devices as well as simulated AFATDS devices, depending on the exercise requirements.

The secondary benefit of adding a STEP interface to COBWebS is that the Cyber Role Player can now use COBWebS to manipulate attributes on entities inside of the simulation. For example, using the STEP interface, COBWebS can now manipulate cyber-related attributes such as CPU-load and memory-load on the tactical devices inside of the simulation. When COBWebS sets the CPU-load to 100% on a tactical device inside of the simulation, that tactical device stops transmitting data. This gives COBWebS the ability to launch cyber-attacks on entities inside of the simulation, instead of only manipulating tactical messages external to the simulation. And by generating cyber-attacks directly on the entities inside the simulation, other entities inside the simulation are now aware of, and can now react to the cyber-attacks.

Potential future work could include targeting other tactical message types that are part of artillery fire missions, to introduce additional CEMA attack effects. For example, currently, only CFF tactical messages are being targeted, but Logistics Report messages could be intercepted to manipulate the types of munitions a gun advertises that is has available or Personnel Status Report messages and manipulate the locations of the guns, which would cause the firing solutions to miss their target when the azimuth and power values are calculated with incorrect gun coordinates.

CONCLUSION

The enhancements made to COBWebS for this AMSO project directly address the technology gap identified by AMSO M&S-enabled communities in 2017, by allowing them to stimulate MCIS with additional and higher fidelity CEMA effects to impact situational awareness and understanding, and by allowing for CEMA actions and effects in the Army's fire mission messaging chain. We were able to accomplish this by incorporating and reusing previously funded technologies, something that is directly in line with AMSO's goal of promoting M&S interoperability and reuse instead of creating single-use solutions. Additionally, the foundation is now in place to expand this capability set even further in the future as the U.S. Army continues to prepare for dominance in the cyberspace warfighting domain.

REFERENCES

Granados, A. (2016, March 23). *Conversion of signal strength in dBm to percentage in WiFi Explorer*. Retrieved from Adrian Granados: https://www.adriangranados.com/blog/dbm-to-percent-conversion

Headquarters, Department of the Army. (1991, July 16). Tactics, Techniques, and Procedures for Observed Fire. Washington, DC, United States of America.

Naval Research Laboratory. (2018, May 18). Builder 3.3 User Manual. Washington, DC, United States of America.

Remcom, I. (2019). *EMPIRE Suite of Propagation Models*. Retrieved from Remcom: https://www.remcom.com/consulting-empire