Distributed Wargaming Visualization and Data Collection

MAJ William Kirschenman
The Research and Analysis Center (TRAC)
1400 Martin Luther King Blvd
White Sands Missile Range, NM 88002-5502
Phone: (575) 678-3389
FAX: (575) 678-6887
william.k.kirschenman.mil@army.mil

AORS Best Presentation/Paper 2021
Working Group 8: Wargaming
9 March 2022

Abstract

The Research and Analysis Center planned and executed the 2020 European Command Capstone Operational Wargame to inform decisions concerning the Army of 2030, the Calibrated Force Posture and the 31+4 Army Futures Command modernization priorities. Corona Virus Disease-2019 restrictions presented resource challenges and prevented bringing approximately 80 subject matter experts and role players to participate in on-site execution. These challenges forced the team to relook traditional methods, models, and tools (MMT) used to conduct wargaming and data collection events in general, and specifically to support a distributed wargame in a classified environment.

The lessons learned during both this and multiple follow-on events have led to the evolution of MMTs to better support both wargames and data collection events focused on analytic efforts.

This paper discusses the critical elements of analytic wargame design and the MMT used to support both the wargame and follow-on events, with a focus on the development and evolution of the Visualization Application and Data Collection in R tool, which was specifically developed to provide distributed role players an appropriate planning and data collection tool.

Introduction

In support of the Army Futures Command (AFC) Capstone Wargame Series, The Research and Analysis Center (TRAC) conducted a wargame in late 2020 focused on the United States (U.S.) European Command (EUCOM) Area of Responsibility (AOR). The purpose of this event was to assess the effectiveness of a prescribed calibrated force posture and identify how various modernization efforts might impact a future fight. The wargame was initially planned as an in-person event since the tools available at the time would only be effective if players were on-site for the event. However, due to stringent Coronavirus Disease 2019 (COVID-19) travel restrictions, TRAC was forced to transition the EUCOM Wargame to a distributed event, with players participating from remote secure internet protocol router (SIPR) workstations across the continental United States. These constraints presented a significant challenge for the wargame team: how do we provide wargame participants a sufficient common operating picture (COP) in order to efficiently and effectively plan their moves and engagements in a classified, geographically distributed environment?

To fill this gap, the Scenarios and Wargaming Division (WWS) at The Research and Analysis Center-White Sands Missile Range (TRAC-WSMR) developed the Visualization Application with Data Collection in R (VADR) tool. This tool provided a sufficient COP in order to allow the EUCOM wargame to be successful. Since then, VADR has continued to evolve. VADR has been used to support a wide variety of data collection events over the past 18 months, to include multiple operational wargames, table top exercises (TTX), rehearsal of concept (ROC) drills, and map exercises (MAPEX). Combat model integrators are able to effectively instantiate detailed study scenarios into multiple combat simulations (One Semi-Automated Forces (OneSAF), Combined Arms Analysis Tool for the 21st Century (COMBATXXI), or Logistics Battle Command (LBC)) due to the adaptability of the VADR output. This paper provides an overview of the initial EUCOM wargame problem, the approach and methodologies used to address this problem, the resulting solution, and an overview of VADR's evolution into a dependable visualization and data collection capability for local or distributed analytical events.

Literature Review

Critical Elements of Analytic Wargame Design

To begin addressing the problem, the team used the four critical elements of analytic wargame design (figure 1) to consider planning factors for a geographically distributed wargame.[1] Most planning factors center around the players and ensuring they can interact effectively with the other critical elements. This wargame required subject matter experts (SME) in their Warfighting functions (WfF) or branches with respect to future concepts and capabilities.

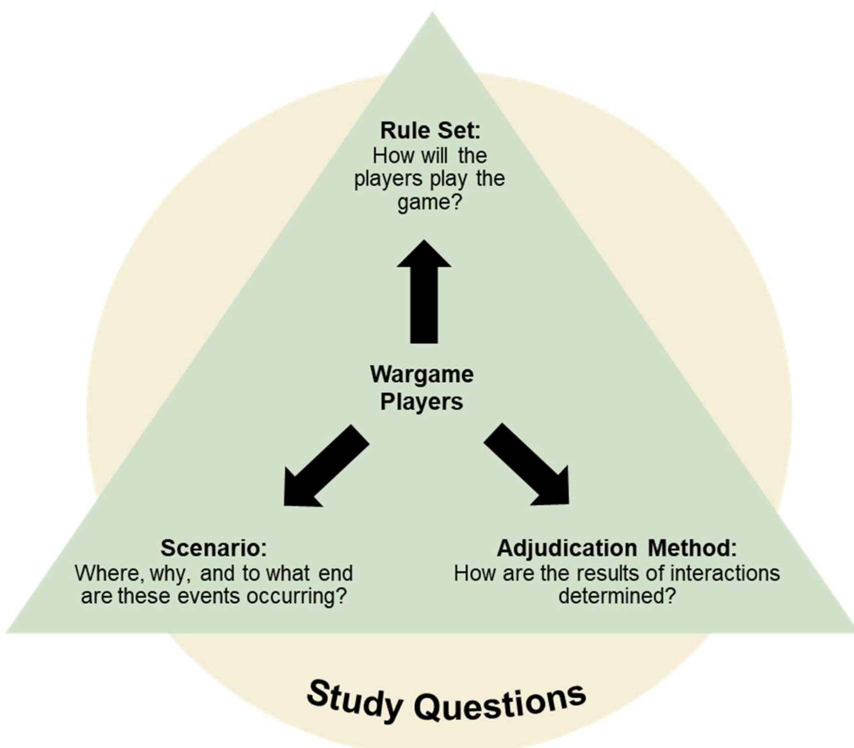

Figure 1. The Four Critical Elements of Analytic Wargame Design.

The players needed an appropriate study scenario to understand background and provide context for the wargame. The scenario information provided players with a road to war to explain the events leading up to the wargame's starting conditions. Subsequently, the study scenario described the wargame's overall objectives and detailed information about resources available to all participants. This information helped put players in the right mindset prior to the wargame, while keeping them on a specific path during execution to help the wargame team collect critical information necessary to answer its study questions.

In order to determine the outcome of player interactions, the wargame team must establish an appropriate adjudication method. TRAC traditionally conducts seminar games at the operational level of warfare which involve discussions between players and SMEs, ending in an agreement for a course of action (COA). However, these seminar wargames also include aspects of system wargames which "rely on a predetermined mathematical method typically based on probabilities" to determine the COA outcomes, and the adjudication method must be crafted around this concept.[2] Given that the wargame scenario time period started more than a decade in the future, employing capabilities that do not exist and an Army Operating Concept of Multi-Domain Operations (MDO) that is currently being experimented with, robust White Cell processes need to be established. These processes need to balance mathematical adjudication results and human adjustments to provide a credible outcome for players while furthering wargame study purposes, all while remaining flexible for efficient implementation regardless of the wargame's setting.

While many planning factors remain consistent, regardless of the wargame's setting, the rule set is impacted by location. The rule set dictates the overall structure of the game and how the players will play it. Since analytic wargame design centers around the players, wargame success is incumbent on providing players with the information they need so they can, in turn, provide the wargame team with data to analyze, synthesize, and gain insights from, to answer the study questions. Many of the methods, models, and tools (MMT) used in traditional in-person wargames are tied to physical processes such as printed information and manuals as references, analog maps and physical game pieces for planning, and heavy desktop computer applications with "operators" or "pucksters". These operators move unit icons, create engagements, and adjudicate in the simulation – all within the confines of a system game. Executing under this paradigm requires extensive resources, including facility reservations, travel arrangements, special information technology (IT) planning and setup, and additional support personnel. The resource burden on the wargame team would naturally decrease in a geographically distributed wargame, putting more emphasis on the wargame players at their respective locations. However, MMT would require significant planning and creativity to effectively design and execute a rule set for distributed participants.

Methods, Models, and Tools

At the start of the EUCOM wargame planning, the wargame team surveyed numerous "in-house" wargaming tools to account for the critical elements of wargame design. The primary tools available included the Wargame Analytical Repository Network (WARnet) and the Versatile Assessment Simulation Tool (VAST) – both developed at The Research and Analysis Center-Fort Leavenworth (TRAC-FLVN). Although these tools could assist in supporting a geographically distributed wargame, the wargame team identified gaps in the rule set.

WARnet is a suite of cloud-hosted tools using Microsoft SharePoint as a foundation service, to enable wargame participants and study analysts to share information, collect data, and observe real-time trends. WARnet supports the scenario and rule set legs of the critical wargame design elements in a way that makes geographically distributed wargames feasible. It acts as a repository for the scenario products that provide wargame background and context, as well as a method to collect player inputs, such as specific unit movements, force-on-force engagements, or qualitative information about player actions in each turn. Once built, WARnet only requires a participant list to grant user accounts, and a brief training session during a wargame ROC drill to ensure participants know how to use the system effectively.

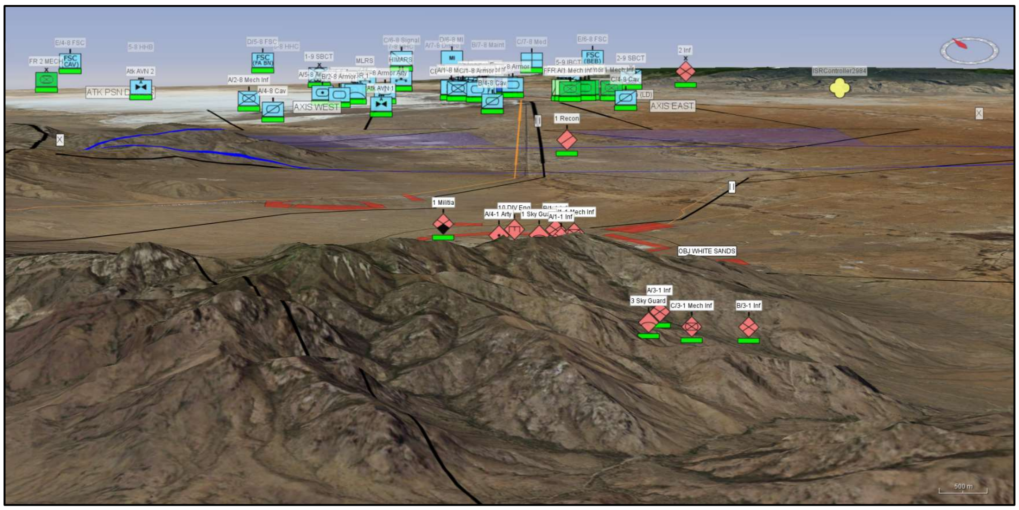

Figure 2. VAST.

VAST is a Java-based wargaming tool that is used for scenario development, concept of operations (CONOPS) development, and study support. It includes a dynamic map interface to plot units and graphic control measures (figure 2). VAST also has a robust backend to build force structures and study scenarios, while collecting data and adjudicating wargame turns by way of modular add-ons, such as different versions of attrition, sustainment activities, or sister service actions. VAST has been the primary method of supporting all three legs of analytic wargame design in traditional in-person games. However, VAST is a computer desktop application with no method of deploying in a cloud-based environment to support a geographically distributed wargame. It is also not feasible to rapidly train participants to use VAST or to send VAST operators to every location to collect wargame inputs for compiled adjudication with the White Cell.

In a geographically distributed setting, the wargame team could use VAST to capture dozens of screenshots during each wargame turn, of different map resolutions and unit filter combinations (normally over 1,000 total units in an operational wargame), and upload them to WARnet for wargame participants to use. This is a potentially feasible solution for visualization and planning, but may not be acceptable or suitable when considering the requirement of providing a sufficient COP to properly inform wargame players, who, in turn, must provide appropriate feedback and input movement or engagement data in a timely manner.

Surveying available wargaming tools highlighted an important gap in the ability to provide an interactive COP to wargame participants. Instead of a VAST operator and screen available for each of the planning cells on-site during an in-person wargame, in a distributed environment, every participant requires access to a cloud-based map product. This map product is critical to facilitate player understanding of unit disposition across the battlefield, while providing sufficient information and context for players to input moves and engagements through WARnet SharePoint forms.

Constraints, Limitations, and Assumptions

Given detailed thoughts about the critical elements of wargame design and the MMTs to address those elements, specific constraints, limitations, and assumptions (CLA) influence the wargame team's solution to its problem of providing a sufficient COP to efficiently and effectively plan moves and engagements in a classified, geographically-distributed environment. COVID-19 restrictions were the largest overall determining factor for the development of the EUCOM wargame's map tool, presenting a variety of trickle-down effects.

The limitations were set by the level of tool development expertise within the wargame team, lack of personnel resources, and no ability to host a web application on a classified network. Previous experience and exposure to the leaflet package in R helped the wargame team make some assumptions as to how the tool should be constructed, while synchronizing existing tool gaps with wargame rule set execution helped determine the features the tool should incorporate. The constraints and assumptions for the tool, influenced by limitations of the wargame team, became the objectives for what features the tool needed to incorporate.

Constraints:

- Wargame participants must use this tool remotely at their respective geographic locations.
- The tool must be simple to use, providing just enough information for users to do what they need; a tool requiring any extensive training is not feasible.
- Tool must be functional on a classified network.
- Tool must incorporate specific force structures.
- Tool must be flexible to give wargame participants a precise snapshot of specific information they may need.
- Tool must have common data standards with WARnet and VAST.
- Tool must act as a viable COP for all wargame participants.

Limitations:

- Wargame team has little software or web application development experience.
- The only flexible programming capability available is the R programming language. TRAC-WSMR computers come pre-installed with the R programming language, RStudio integrated development environment (IDE), and ability to install additional packages as required. No equivalent programming solution exists for the wargame team.
- No available classified RStudio Connect Server (RSC) to host a web application.[3]
- Wargame team lacks personnel to operate a tool for wargame participants.

Assumptions:

- Static maps screenshots are not a suitable option for executing the wargame rule set.
- Tool should be lightweight for hosting flexibility.
- The R programming language will be used to create the tool based on wargame team programming experience.
- Tool should incorporate dynamic maps with ability to zoom in and out with a heavy reliance on the leaflet JavaScript library based on wargame team programming experience.[4]
- Tool should include additional, relevant information to facilitate wargame player planning and inputs of turn moves and engagements in a timely manner.
- The tool needs the ability to rapidly be updated and redeployed whenever changes need to be made (coinciding with wargame turns or other updates).

Methodology

Since this tool needs to be used by wargame participants to filter and see unique information, the wargame team looked at this specific objective through the lens of the players, as depicted in figure 3. The wargame team needs to deploy a tool that has enough information to be valuable to all participants, while being customizable to the requirements of any individual. For example, a player representing the Fires WfF might need to differentiate between fires assets at different echelons while filtering Threat units to specific types of targets in order to compare Friendly fires system ranges with those of the Threat units.

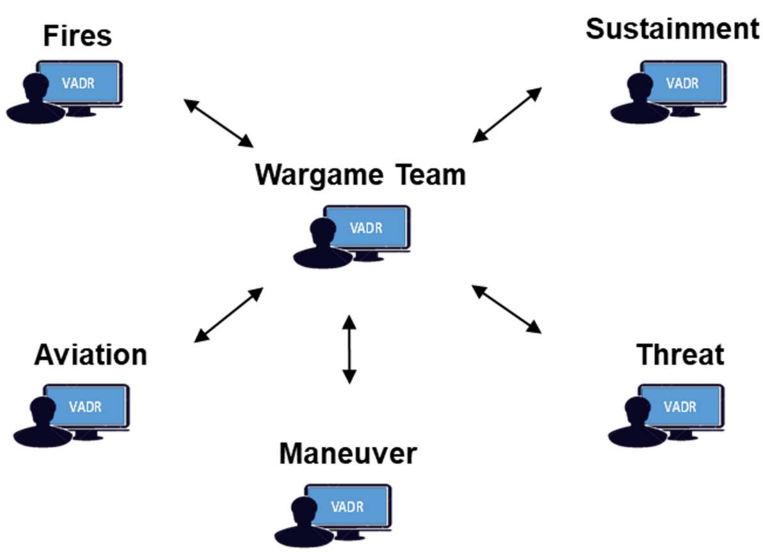

Figure 3. Notional Representation of Wargame Participants/WfFs.

Alternatively, a player representing the Sustainment WfF may need to see all maneuver and sustainment units, trafficable bridges, and supply routes in order to effectively sustain supported units as they maneuver through turns. A wargame player representing U.S. Army Aviation may need to see both combat radius and weapon range rings, as well as Threat surface-to-air assets, to determine what flight routes could be taken during a wargame turn and what critical targets he may be able to engage.

Combining these types of considerations with the list of CLAs, the wargame team determined that programming a map tool in R produces the most feasible solution for answering the overarching problem induced by conducting a classified, geographically-distributed wargame. The R programming language, RStudio IDE, and common supporting packages come with standard computer images across all networks at TRAC-WSMR, with the ability to easily install additional packages as needed. With no RSC available to deploy a web application using the "Shiny" package in R, the wargame team's solution needs to be housed entirely in a hypertext markup language (HTML) file to be hosted on WARnet.[5] Wargame team experience levels with both R and the leaflet package were based on graduate school programming, specific instruction in the Operations Research Systems Analysis Military Applications Course (ORSA MAC), "Intro to R" course materials from the Center for Army Analysis (CAA), or personal investigation.

A base leaflet map was created by piping (sequencing multiple operations) standard, unchanging features into `leaflet::leaflet()` such as adding sets of map tiles, setting the default view, adding a drawing toolbar and distance measurement tool, a scale bar, and a layer control feature for cycling through different base maps or unit filters that will subsequently be created.[6] Map tile types, zoom levels, and boundaries were planned and determined in advance, then downloaded on an unclassified system to be put into a folder structure on WARnet. This map tile process was extremely tedious due to large file downloads, network transfer times, and map tile organization. A better map tile solution involving Web Map Services (WMS) will be discussed in the Insights section.[7]

Next, to meet several of the tool objectives including "common data standards with WARnet and VAST" and "incorporate specific force structures," the wargame team needed to create a force structure workbook to synchronize data outputs of VAST or WARnet, giving the ability to render and provide appropriate information in this tool. A typical VAST entity output file includes all information required for rendering unit icons in its display, along with information used to supplement engagement data for adjudication. Most rendering is done in the internal workings of VAST, so the wargame team needed to be able to import the VAST entity output file, while joining it with a detailed workbook that helped build icons on the map and supplemental information in the map tool. Table 1 shows a notional example of this synchronization between VAST and VADR. A VAST entity output file includes information not limited to the first four columns in orange and green, while the VADR workbook includes information in the last five columns in green and yellow. The "vast_type" column was used as the primary key to join the two files, resulting in the appropriate information needed to start piping more commands into the base leaflet map.

Table 1. Synchronization Between VAST Output and VADR with Notional Data.

unit_side	strength	vast_name	**vast_type**	unit_size	unit_function	range	notes
Friendly	90	1-2 IN BN	U.S. IN BN	battalion	Maneuver	5	dismounts
Friendly	85	3-4 AVN CO	U.S. AVN CO	company	Aviation	150	1x Shadow
Threat	80	5 ADA BDE	EN ADA BDE	brigade	Air Defense	100	3x BN
ADA – air defense artillery *AVN – aviation* *BDE – brigade* *BN – battalion*				*CO – company* *EN –engineer* *IN – infantry*			

With a cleaned data frame of information loaded in RStudio, along with a custom-built set of image files corresponding to the types of units represented in the EUCOM Wargame force structure, the team completed the following steps to add and pipe additional features to the base leaflet map:

- Unit filter lists were created based on the "unit_side" attribute alone (an overall "Friendly" and "Threat" filter), or the composite keys of the "unit_side" and "unit_function", producing "Friendly Maneuver" as an example for the first row in Table 1. These unit filters were then added to the overlay list for the base leaflet map.
- Unit icons were built under these unit filters using the `leaflet::icons()` function and the "vast_type", "unit_size", and "strength" attributes to render the type of unit, the echelon, and the strength bar respectively.
- Units were then rendered on the base leaflet map using the `leaflet::addMarkers()` function with these newly created icons, adding a popup label that displays attributes like "vast_name", "strength", "range" and definition for that range (e.g., weapon system range versus radar detection range), "parent_unit" to filter WARnet movement and engagement inputs, and "notes" to describe additional information that wargame participants may need to know about that unit.
- Range rings were built to display under the composite key filters using `leaflet::addCircles()` and the "range" attribute, to display respective unit range rings. These range rings equate to movement for Maneuver units, combat radius for Aviation units, or weapon system ranges for Fires units.

Once all of these features were piped to the base leaflet map, the completed leaflet map was saved to an HTML file using the `htmlwidgets::saveWidget()` function.[8] This step allowed the interactive map tool to be hosted on the WARnet dashboard site for the wargame either by embedding it onto the HTML page or providing a link to open it in a different browser tab. The force structure workbook and code were in a state where upon wargame turn completion and a new VAST entity output file, the VADR code could immediately be executed to generate an updated HTML file for WARnet. The solution for the wargame team's problem had come to fruition and was initially called Visualization Augmentation Distributed in R, or VADR (the acronym definition changed as the tool later evolved to also provide data collection capability, which will be discussed in the Insights section). VADR provided wargame participants with a sufficient COP in order to efficiently and effectively plan their moves and engagements in a classified, geographically-distributed environment.

Results

During wargame execution, participants had an accurate and frequently updated COP at their disposal. An example of the VADR user interface (UI) is shown in figure 4 and is based on an interactive map (e.g., Google Maps) with the ability to move the map to any location – independent of other participants – and scroll to increase or decrease resolution. Users were able to see any combination of filtered units they would like, at the desired resolution, to help them effectively plan in a geographically-distributed environment during wargame turns. The drawing toolbar added the ability to draw shapes on the map as graphic control measures, while formatting those shapes and taking screenshots for direct input into turn action slides. These turn action slides provided additional context for White Cell adjudication and acted as a historical artifact for the wargame.

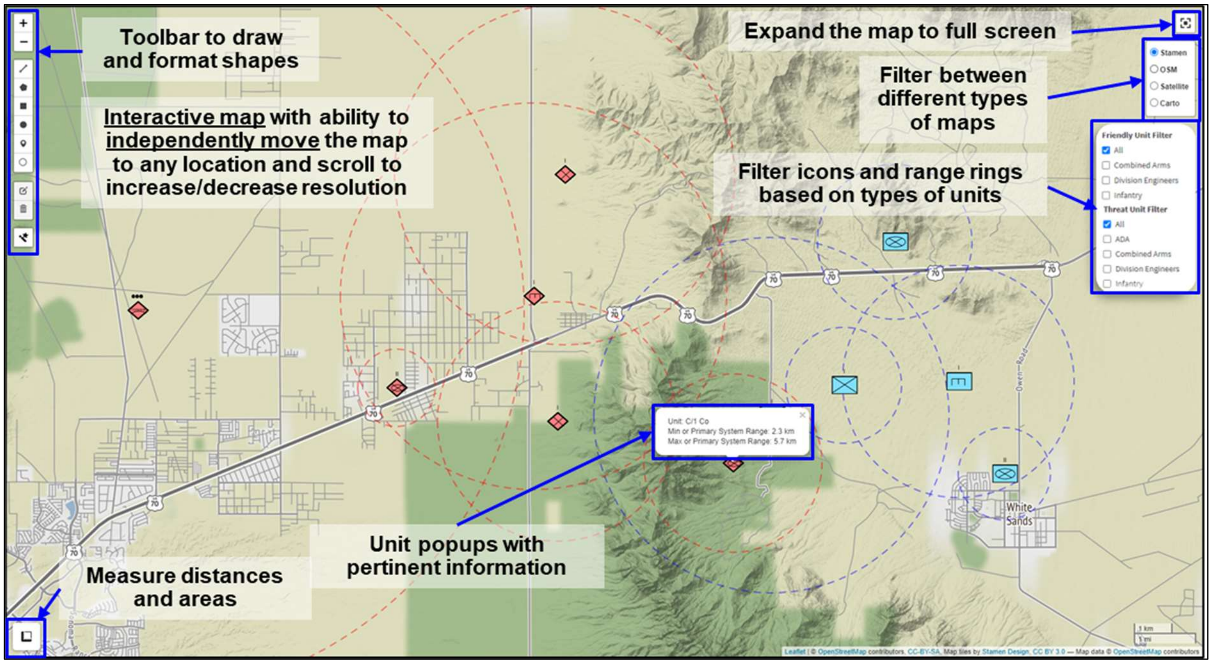

Figure 4. Example Wargame VADR User Interface with Notional Data.

Wargame players used VADR's features to help coordinate plans with the Higher Headquarters and across the different WfFs and domains, and then used the tool to determine what moves or engagements they needed to make. The data collection method encompassed inputting move and engagement data into WARnet. For unit moves, a player identified the unit to move and then used the distance measurement tool to find the coordinates of the new location. The player then used information in the unit popup, such as parent unit and unit name, to filter down and input information for that move. For engagements, a player focused on one or more units involved in an engagement, both on the Friendly and Threat sides, then filtered, in a similar way as was done in unit moves, to set up the engagement. Finally, the player determined the type of attack and submitted the engagement. Once all players input their wargame turn moves and engagements from their remote locations, this data was exported from WARnet into VAST, where all moves and engagements were adjudicated through VAST's attrition models and then reviewed by the White Cell. Once the adjudication portion of the wargame rule set was complete, the process started over with a new VAST entity output file post-adjudication, followed by the deployment of a new VADR snapshot to start planning in the next turn. This process worked flawlessly throughout the wargame and provided shared understanding of the COP across all wargame participants.

Insights

VADR successfully provided an interactive COP for classified, geographically-distributed wargames, filling a gap of which there was no equivalent solution. Many computer desktop applications exist either "in-house" such as VAST, or external to TRAC, that incorporate all of the capabilities of VADR and more. The primary attribute that distinguished VADR from the others was its ability to be hosted in the cloud (WARnet), easily used by all participants at their remote locations. Providing this capability would have otherwise been impossible given the constraints the wargame team faced and the absence of software that could perform equally under given constraints. The success of VADR precipitated further tool experimentation and development during future events.

Visualization and Data Collection

Following the conclusion of the EUCOM Wargame, WWS developed a data collection management plan and executed a MAPEX for the TRAC-WSMR Studies and Analysis 2 Directorate's Optionally Manned Fighting Vehicle (OMFV) Phase II Study scenario data collection efforts. In order to address the needs of the Next Generation Combat Vehicle Cross (NGCV) Functional Team (CFT), the study team needed to develop a defensive scenario for instantiation into OneSAF that would allow them to analyze outcomes for a future fight to address specific study questions. OneSAF integrators required detailed information about units by turn or phase such as task, purpose, and location, as well as any number of behaviors including movement formations, engagement criteria, or decision points.

The primary issue facing WWS and the study team involved how to execute the MAPEX under similar COVID-19 constraints experienced during the EUCOM Wargame. WWS was still working under the limitation of no available classified RSC, which was still required to host a Shiny application. In lieu of RSC, since it was ultimately determined that this OMFV Phase II Study MAPEX would be conducted entirely in person at TRAC-WSMR, using a Shiny application was now an option for evolving VADR into a visualization and data collection tool. Since all TRAC-WSMR computers come with R and RStudio installed, it was straightforward to create a Shiny application, and put all application and dependent files in shared drive directories accessible to all participants.

Building upon the VADR tool that was used purely for visualization in the EUCOM Wargame, the source code was modified and expanded on to make UI customizability virtually boundless, introduce reactivity to have any variety of backend tasks occur, and allow for persistent data storage. The main R package that allowed this drastic refactoring was Shiny. Shiny allowed WWS developers to alter features hard coded into the EUCOM Wargame tool's UI such as unit filters and range rings, separating them from the map, as figure 5 illustrates.

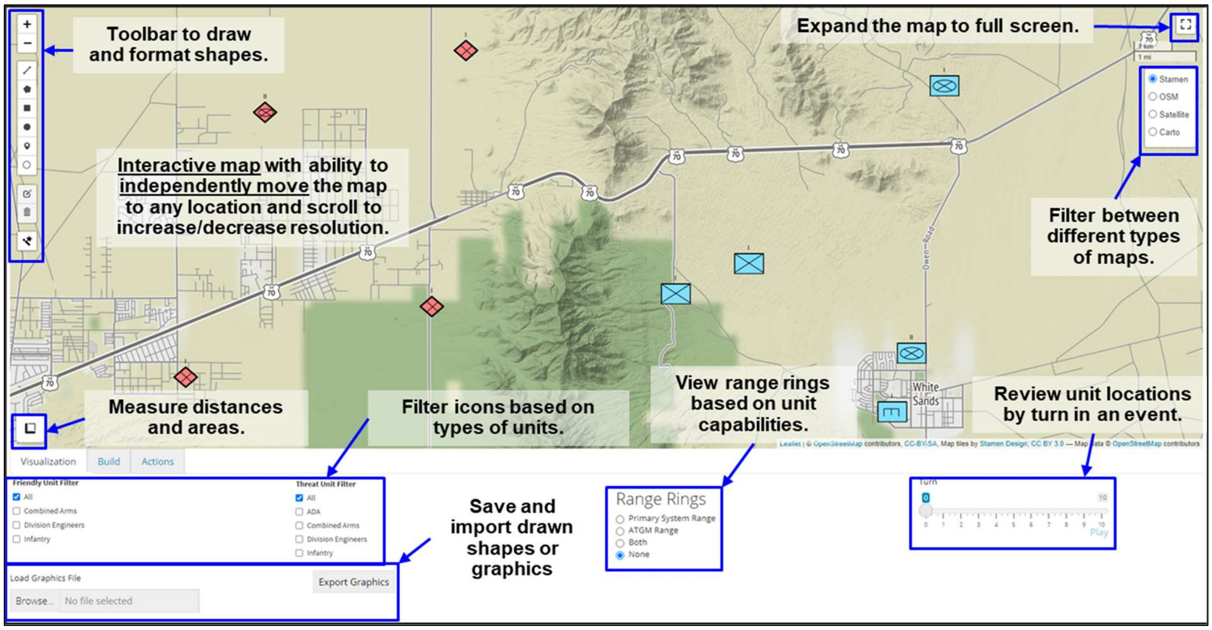

Figure 5. Example OMFV Phase II MAPEX VADR (Visualization) User Interface with Notional Data.

Multiple turns' worth of unit information is now accessible in VADR through the use of a turn slider input. Additionally, shapes can be drawn with the drawing tool bar, saved, and imported into another VADR user's instance of the application. Unit icons are also now moveable, allowing users to move icons across the screen, and capture specific geospatial data. All user interactions with the UI can also be made reactive, triggering additional actions to occur.

Outside of the reactive UI objects seen in figure 5 that trigger actions to occur, the dragging and dropping of unit icons on the leaflet map allowed for data collection and persistent data storage. When players moved unit icons, they triggered VADR to save the old and new locations of that icon and to open a data collection dialog prompting the user to input specific information required for OneSAF scenario integration as shown in figure 6. The user then confirmed the unit move or reverted it back to its original location by closing the window. Otherwise, specific data was collected for that unit, submitted, and reviewed in the "Actions" tab at the bottom of the UI. Finally, using the "DT" package in R (an interface to the DataTables JavaScript library), the collected data was reviewed and adjusted as needed, ensuring the study team and OneSAF integrators had all of the information needed for that turn.[9]

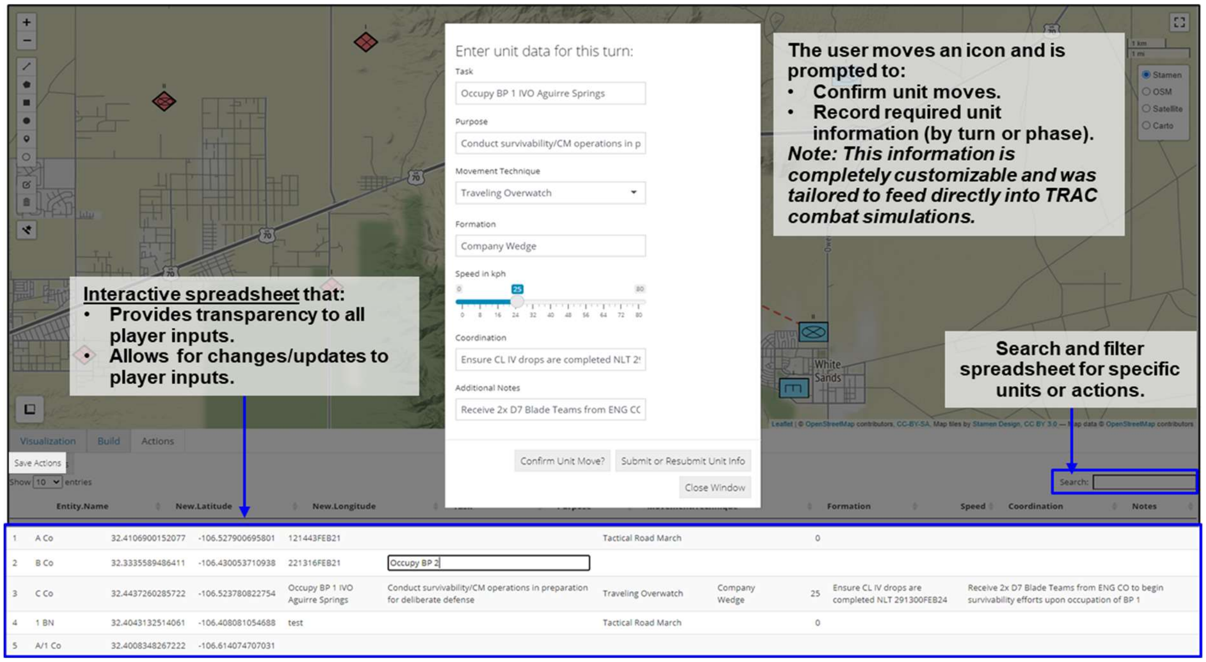

Figure 6. Example OMFV Phase II MAPEX VADR (Data Collection) User Interface with Notional Data.

With VADR now able to be used as a Shiny application with both visualization and data collection, the MAPEX rule set could be executed on-site at TRAC-WSMR, with multiple planning cells, reducing the need for analog maps and map pieces, or a team of data collectors hand writing notes for each player. MAPEX participants included members of the NGCV CFT executing Friendly forces planning, the AFC Directorate of Intelligence and Security (DoIS) executing Threat forces planning, and TRAC-WSMR OneSAF integrators, all synchronized and working together. These player-integrator teams planned and recorded the required information in VADR that helped instantiate the defensive study scenario into OneSAF within a month, ready for test runs to commence. An effort that previously took several months to accomplish. Previous methods of conducting MAPEXs relied on increased resources and analog products, creating a lag in time post-MAPEX to compile all of the data before OneSAF integration could begin – a lag that typically lasted several months. Using this Shiny version of VADR to conduct the OMFV Phase II MAPEX allowed for deliberate data collection based on study team needs, with complete organization of actions to easily use as a guide for integration, while allowing for direct imports of all geospatial data collected.

Map Tiles

The method for utilizing map tiles during the EUCOM Wargame and OMFV Phase II MAPEX involved careful planning of selected base maps, bounding boxes, and appropriate zoom levels to ensure the event participants could understand the COP and plan. Since the EUCOM Wargame was fought at the operational level with Corps-level units down to Brigades (some units were fought at Battalion or Battery level, such as Fires units), deep zoom levels were not needed. Table 2 shows zoom levels with equivalent rendered size of one respective tile at that zoom level.[10] Traditional map tiles are stored in a quad tree data structure based on Equation (1).[11] This equation determines the number of tiles that comprise a specific zoom level that when knitted together encompasses the entire Earth. For example, at a zoom level of 0, one tile encompasses the entire Earth, whereas a zoom level of 3 (one tile equal to the size of a continent) would require 64 tiles to cover the Earth. This equation needs to be adjusted and calculated for every zoom level required, which helps understand the precise number of tiles that were needed to be hosted on the WARnet SharePoint site or the TRAC-WSMR shared drive for each of these respective events.

Table 2. Single Map Tile Visual Representation at Different Zoom Levels.

At Zoom Level	You Can See
0	The Earth
3	A Continent
4	Large Islands
6	Large Rivers
10	Large Roads
15	Buildings

$$n_{tilesZoomLevel} = 2^{zoom} \cdot 2^{zoom} \tag{1}$$

The EUCOM Wargame required expansive coverage at higher zoom levels to understand conditions outside of the EUCOM AOR, so map tiles of the entire Earth were used for zoom levels 0-7, while zoom level 8-13 map tiles were bounded by the land area of operations inside the combined Joint operations area. Based on different file sizes of 256 x 256-pixel map tile images, each dependent on the base map type, using three different base maps to give users a variety of options equated to approximately 1.75 gigabytes (GB) of map tiles. This put a significant amount of strain on the WARnet SharePoint site, while also adding tedious tasks to plan, prepare, download, and transfer these files appropriately from unclassified to classified systems. The OMFV Phase II MAPEX required an even larger number of tiles at deeper zoom levels due to the tactical nature of the operation, equating to 3 GB of data. The map tiles for the OMFV Phase II MAPEX were easier to manage as they were placed in a local shared drive folder structure, but the tedious steps prior to this transfer remained the same.

The next series of data collection events WWS supported involved the AFC Standard Scenario 2.x development effort. Through continuous research, the team was able to take advantage of WMS map tiles, in similar fashion to how Geospatial Information System (GIS) software uses base map layer connections or certain classified browser-based interactive map solutions like Google Maps function.[12]. The National Geospatial-Intelligence Agency (NGA) has a classified WMS server with a wide variety of base map layers. Utilizing NGA WMS base maps reduced the amount of time needed to plan data collection events, and also made the VADR (visualization only) HTML tool standalone and functional without needing downloaded map tiles as long as the respective WMS server could be accessed through a network connection. VADR was successfully deployed and utilized during the AFC Standard Scenario 2.x CONOPS Development Wargame from August to September 2021, in a significantly quicker and lighter fashion using a WMS server. This provided an alternative solution to the tedious process of downloading, transferring, and burdening the WARnet site with large amounts of map tiles.

Interactive HTML Dashboards

Coinciding with the AFC Standard Scenario 2.x Wargame, WWS developed a solution for incorporating "Shiny-like" reactivity into an HTML dashboard. Instead of saving a singular HTML widget as had been previously done for wargames, this new solution allowed for knitting multiple HTML widgets into one dashboard saved as a singular HTML file. To do this, the team used the flexdashboard and RMarkdown packages in R.[13] Flexdashboard allowed for the grouping of different HTML widgets within a dashboard construct, while RMarkdown knitted the UI together as a standalone HTML file. Much like narrative markdown text and code chunks can create high quality documents, reports or presentations, RMarkdown can do the same with dashboards.

These new packages allowed the team to transform the VADR HTML tool previously described in figure 4 into the knitted flexdashboard RMarkdown version shown in figure 7. The new appearance and underlying structure enabled a more professional UI, with the ability to add tutorial documents and support references. Future expansion with flexdashboard could include additional tabs showing previous wargame turns, or the inclusion of a number of other HTML widgets that are compatible with the crosstalk package in R (such as interactive spreadsheets and plots).[14]

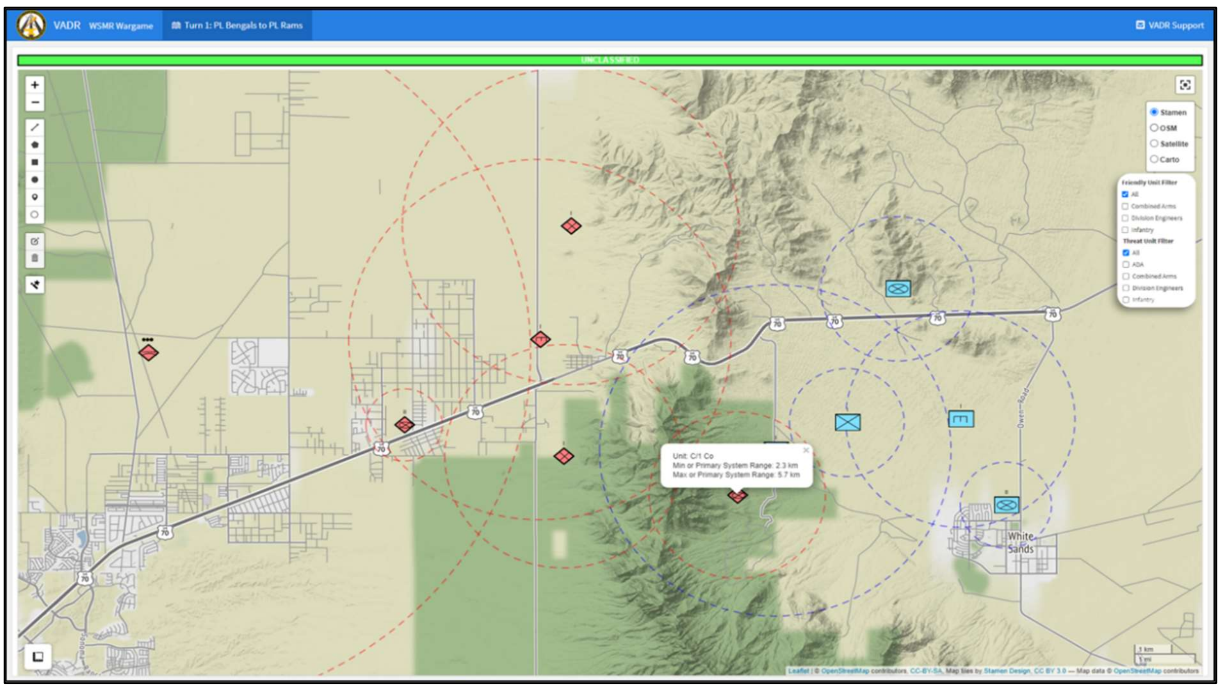

Figure 7. Example VADR User Interface using Flexdashboard with Notional Data.

Code Development, Sustainment, and Sharing

Prior to adding a data collection capability to VADR for the OMFV Phase II MAPEX, WWS used the local shared drive to store and make updates to the code, relying on word-of-mouth confirmations and trust without any version control system in place. After the MAPEX, the team shifted to using Git as a distributed version control system and a GitLab website to host a Git remote repository as seen in figure 8.[15] This resulted in more streamlined and efficient code development since team members could either use the Git workflow directly within their RStudio UI with the TRAC-WSMR approved Git for Windows, or use the git2r package in lieu of local approval for Git for Windows. Hosting the main VADR repository on GitLab allowed the team to manage project permissions, use Kanban boards for agile development, and provide an information sharing platform for this tool. Additionally, GitLab allows other TRAC analysts to have access to the source code, read a tutorial or overview in the README file, or fork the project and tailor it for personal use if desired.[16]

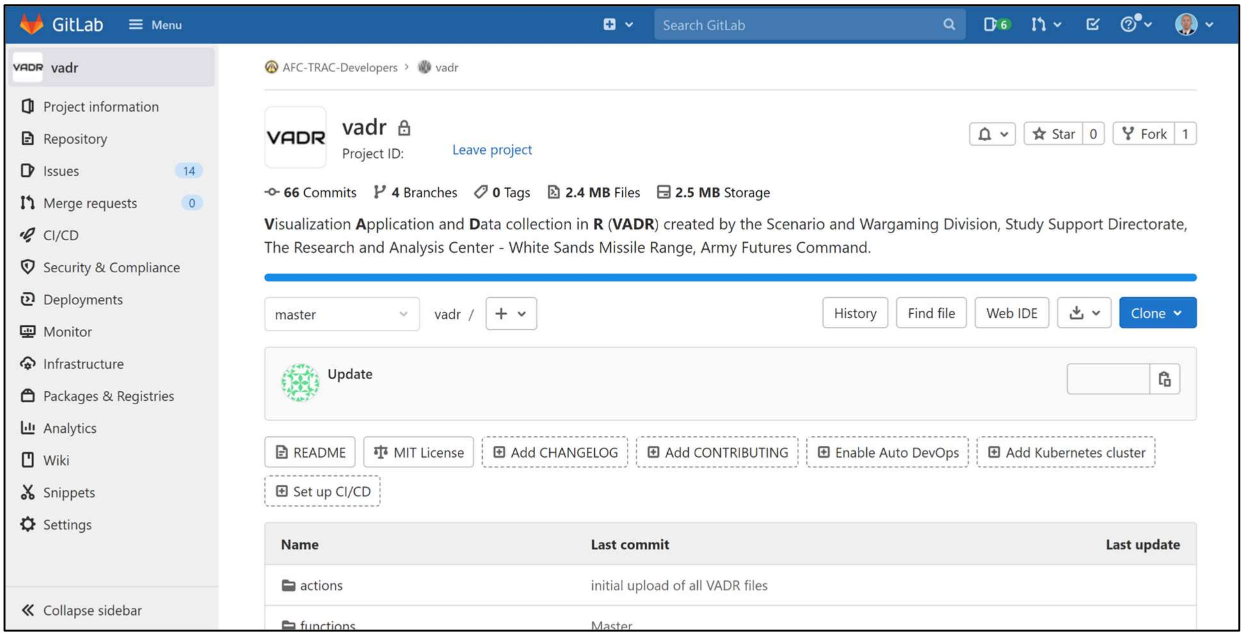

Figure 8. VADR Project on GitLab.

WWS hopes to spread the idea of sustainable code development and sharing through this GitLab effort, with the purpose of sharing valuable tools across the TRAC community in order to prevent constant recreations of previously developed efforts. This would allow analysts to survey what tools are available in the community, while forking a specific project that may be of use to them in order alter backend (or frontend) functionality to contribute to their specific study or analytic effort. Additionally, collaboration and feedback to and from project owners will make the base tool better or assist the analyst forking the tool for their own reasons. WWS has currently settled on a Department of Defense IL5 GitLab site which allows for code and data storage up to high sensitivity controlled unclassified information.

Conclusions

VADR has become more of a capability for WWS since it provides a range of base tool options that can easily be customized under the hood for specific efforts. It has constantly adapted over time to support the mission, whatever that mission may be. VADR initially started as a tool to fill a wargaming visualization gap in classified, geographically-distributed environments, evolving into a Shiny application with added data collection capability and a slew of other features as a web application. Another fork off of the original VADR (HTML for visualization only) became a structurally different tool, providing even more functionality in an HTML file, and pushing the bounds of what the team knew a stand-alone HTML file could do.

Since its inception, VADR has supported the visualization and/or data collection efforts of a wide variety of events, from distributed wargames, to local MAPEXs executed to instantiate a detailed study scenario into combat simulations, to ROC drills for rehearsing Joint CONOPS. The VADR source code is also resident at the Division level for use by the Division ORSAs for their own use in command post exercises or wargames. VADR demonstrates an ability to customize a tool's UI for visualization, planning, and data collection. It is inherently designed to be tailored and used by analysts, flexible to adapt to study timelines through a "just enough" design philosophy. It enhances data collection and processing through real-time, interactive, and accessible collection methods, while allowing for a more efficient allocation of resources. It is also flexible to operate on both classified and unclassified networks, within on-site, partially distributed, and fully distributed environments. The VADR capability has been greatly beneficial to TRAC and the supported community and will continue to serve as a viable visualization and data-collection solution. VADR also provides an incredible opportunity for junior analysts to dive into the world of R programming and web applications, while benefitting study efforts where equivalent digital solutions may not exist.

Recommendations

With distributed wargaming becoming more of the norm than the exception in the ongoing COVID-19 environment, TRAC is looking at shifting from a dependence on multiple solutions like VADR, VAST, and WARnet in a wargame, to one state-of-the-art wargaming solution that incorporates all the good features of these solutions. The new tool will refactor VAST into the R programming language, adding features from VADR and WARnet, to overcome the challenges of the Risk Management Framework process, operating under the Data Science Environment umbrella.[17] The new tool will be cloud-based, better enabling distributed wargames, local events, or anything in between.

While the appetite for the VADR capability within WWS continues in lieu of other tools that either cannot fill the same gaps or cannot do so under time constraints, significant development resources or efforts will be reserved for assisting with development of TRAC's new wargaming tool, in the hopes of making it fully operationally capable as soon as possible. Until that time, WWS will continue to support wargames and other data collection events with VADR. As for the wider analytic community, as long as analysts are required to build relevant, bespoke, specific, purpose-built models, VADR will always provide a potential starting point as well as a great method for learning about R programming and building Shiny web applications.

List of Acronyms

ADA	air defense artillery
AFC	Army Futures Command
AOR	area of responsibility
AVN	aviation
BDE	brigade
BN	battalion
CAA	Center for Army Analysis
CFT	cross-functional team
CLA	constraints, limitations, and assumptions
CO	company
COA	course of action
COMBATXXI	Combined Arms Analysis Tool for the 21st Century
CONOPS	concept of operations
COP	common operating picture
COVID-19	Coronavirus Disease 2019
DoIS	Department of Intelligence and Security
EN	engineer
EUCOM	European Command
GB	Gigabyte
GIS	Geospatial Information System
HTML	hypertext markup language
IDE	integrated development environment
IT	information technology
LBC	Logistics Battle Command
MAPEX	map exercise
MDO	multi-domain operations
MMT	methods, models, and tools
NGA	National Geospatial-Intelligence Agency
NGCV	Next Generation Combat Vehicle
OMFV	Optionally Manned Fighting Vehicle
OneSAF	One Semi-Automated Forces
ORSA MAC	Operations Research Systems Analysis Military Applications Course
ROC	rehearsal of concept
RSC	RStudio Connect Server
SME	subject matter expert
TRAC	The Research and Analysis Center
TRAC-FLVN	The Research and Analysis Center-Fort Leavenworth
TRAC-WSMR	The Research and Analysis Center-White Sands Missile Range
TTX	table top exercise

U.S.	United States
UI	user interface
VADR (initial)	Visualization Augmentation Distributed in R
VADR (current)	Visualization Application with Data collection in R
VAST	Versatile Assessment Simulation Tool
WARnet	Wargame Analytical Repository network
WfF	Warfighting function
WMS	Web Map Services
WWS	Scenarios and Wargaming Division

Bibliography

Brian Wade, 2018, "Four Critical Elements of Analytic Wargame Design", *Phalanx*.

Peter Perla, 1990, *The Art of Wargaming: A Guide for Professionals and Hobbyists*, Naval Institute Press.

[1] This paper uses the framework described in LTC Brian Wade's article "The Four Critical Elements of Analytic Wargame Design" as published in the Military Operations Research Society's *Phalanx* publication in December 2018 (Vol. 51, No. 4).
[2] Definitions for "seminar" and "system" wargames are taken from Peter Perla's book, *The Art of Wargaming: A Guide for Professionals and Hobbyists*, published on March 1, 1990.
[3] RStudio Connect Server is a standalone publishing platform for the work that teams create in R. It allows users to publish web applications or other content to be hosted in the cloud, for anyone with appropriate permissions to access.
[4] Leaflet is the leading open-source JavaScript library for interactive maps and is designed with simplicity, performance, and usability in mind. There is also a leaflet package in R that includes most of the Leaflet JavaScript capability, with the ability to add Leaflet JavaScript plugins to R code as needed.
[5] Shiny is an R package that makes it easy to build interactive web applications straight from R. It allows for hosting standalone applications on a webpage, embedding them in R Markdown documents, or building dashboards.
[6] Pipes are a powerful tool for clearly expressing a sequence of multiple operations in R. The pipe, "%>%", comes from the magrittr package in R. Magrittr has two aims: decrease development time and improve readability and maintainability of code.
[7] A Web Map Service is a standard protocol developed by the Open Geospatial Consortium in 1999 for serving georeferenced map images over the internet. These images are typically produced by a map server from data provided by a Geospatial Information System database.
[8] Htmlwidgets is a package in R that uses JavaScript visualization libraries at the R console, bringing the ability to embed widgets in R Markdown documents and Shiny web applications, and save them to HTML.
[9] The DT package in R provides an R interface to the JavaScript library, DataTables. This allows for R data objects (matrices or data frames) to be displayed as tables on HTML pages, and DataTables provides filtering, pagination, sorting, and many other features in the tables.
[10] This table corresponds to how mapbox-based map zoom levels function. Mapbox is a company that specializes in powerful developer tools for mapping. The Leaflet library forms the basis of Mapbox.js, a plugin for using Leaflet maps.
[11] This equation is used to calculate the number of individual tiles in zoom level quadtrees and is the equation used by mapbox.
[12] A GIS is a spatial system that creates, manages, analyzes, and maps all types of data. GIS connects data to a map, integrating location data with all types of descriptive information.
[13] Flexdashboard is a package in R that makes it easy to create interactive dashboards for R, using R Markdown. RMarkdown is a package in R that helps create dynamic analysis documents that combine code, rendered output (such as figures), and prose. The user brings data, code, and ideas, and R Markdown renders the content into a polished document used to do data science interactively within RStudio, reproduce analyses, collaborate and share code with others, and communicate results with others.
[14] Crosstalk is a package in R that acts as an add-on to the htmlwidgets package. It extends htmlwidgets with a set of classes, functions, and conventions for implementing cross-widget interactions.
[15] Git is a free and open-source distributed version control system designed to handle everything from small to very large projects with speed and efficiency. Git is easy to learn and has a tiny footprint with lightning-fast performance. It includes features like cheap local branching, convenient staging areas, and multiple workflows. GitLab is a DevOps platform, delivered as a single application that spans the entire software development lifecycle. GitLab started as an open-source project to help teams collaborate on software development. DevOps is the combination of cultural philosophies, practices, and tools that increases an organization's ability to deliver applications and services at high velocity.
[16] A Kanban board is an agile project management tool designed to help visualize work, limit work-in-progress, and maximize efficiency (or flow). Agile project management is an iterative approach to managing software development projects that focuses on continuous releases and incorporating customer feedback with every iteration.
[17] The Risk Management Framework was developed by the National Institute of Standards and Technology and provides a process that integrates security, privacy, and cyber supply chain risk management activities into the system development life cycle. Applications developed and deployed within Data Science Environments are not required to go through the Risk Management Framework process.