NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

INTEROPERABILITY OF NETWORKED SIMULATION SYSTEMS FOR DISTRIBUTED FIRE SUPPORT COORDINATION TRAINING

by

Alexander R. White

June 2021

Thesis Advisor:	Curtis L. Blais
Co-Advisors:	Kirk A. Stork
	Christian R. Fitzpatrick

Research for this thesis was performed at the MOVES Institute.

THIS PAGE INTENTIONALLY LEFT BLANK

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

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.

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

4. TITLE AND SUBTITLE INTEROPERABILITY OF NETWORKED SIMULATION SYSTEMS FOR DISTRIBUTED FIRE SUPPORT COORDINATION TRAINING	**5. FUNDING NUMBERS**
6. AUTHOR(S) Alexander R. White	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) ONR Code 34, Arlington, VA 22203	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

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

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

13. ABSTRACT (maximum 200 words)

Coordination of air and surface fires with maneuver provides an ideal target for simulation training due to the limited availability, high cost, and risk associated with live fire exercises. Current simulation systems provide task-oriented training to operators, but no opportunity to practice communicating and coordinating with other agencies. This thesis uses the Distributed Simulation Engineering and Execution Process to guide the creation of a simulation environment that addresses this training capability gap by demonstrating interoperability of simulations for ground observers, close air support, constructive surface fires, and communication tools in a realistic combined arms scenario. A simulation environment featuring Bohemia Interactive Simulations' VBS4, Lockheed Martin's PREPAR3D, and Battlespace Simulations Inc.'s MACE was developed using the Distributed Interactive Simulation (DIS) standard along with ASTi Voisus communications software. Although a research virtual private network (VPN) was available, DIS broadcast communications could not be supported between VPN clients. The simulation environment was run on a local network and distant users utilized remote desktop connections. Although VBS4 suffered performance issues and PREPAR3D is not ideal for close air support, MACE and ASTi Voisus performed well and the simulation environment was successful. For physically distributed training a High-Level Architecture (HLA) or multi-architecture federation is recommended.

14. SUBJECT TERMS close air support, DIS, distributed, DSEEP, fire support coordination, fires, interoperability, networked, simulation, simulator, training	**15. NUMBER OF PAGES** 119
	16. PRICE CODE

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

NSN 7540-01-280-5500

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

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

INTEROPERABILITY OF NETWORKED SIMULATION SYSTEMS FOR DISTRIBUTED FIRE SUPPORT COORDINATION TRAINING

Alexander R. White
Major, United States Marine Corps
BS, Abilene Christian University, 2009

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MODELING, VIRTUAL ENVIRONMENTS, AND SIMULATION

from the

NAVAL POSTGRADUATE SCHOOL
June 2021

Approved by: Curtis L. Blais
Advisor

Kirk A. Stork
Co-Advisor

Christian R. Fitzpatrick
Co-Advisor

Gurminder Singh
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Coordination of air and surface fires with maneuver provides an ideal target for simulation training due to the limited availability, high cost, and risk associated with live fire exercises. Current simulation systems provide task-oriented training to operators, but no opportunity to practice communicating and coordinating with other agencies. This thesis uses the Distributed Simulation Engineering and Execution Process to guide the creation of a simulation environment that addresses this training capability gap by demonstrating interoperability of simulations for ground observers, close air support, constructive surface fires, and communication tools in a realistic combined arms scenario. A simulation environment featuring Bohemia Interactive Simulations' VBS4, Lockheed Martin's PREPAR3D, and Battlespace Simulations Inc.'s MACE was developed using the Distributed Interactive Simulation (DIS) standard along with ASTi Voisus communications software. Although a research virtual private network (VPN) was available, DIS broadcast communications could not be supported between VPN clients. The simulation environment was run on a local network and distant users utilized remote desktop connections. Although VBS4 suffered performance issues and PREPAR3D is not ideal for close air support, MACE and ASTi Voisus performed well and the simulation environment was successful. For physically distributed training a High-Level Architecture (HLA) or multi-architecture federation is recommended.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

3DOF	3 degrees of freedom
AAR	after-action review
AAV	amphibious assault vehicle
ADVTE	Aviation Distributed Virtual Training Environment
AI	artificial intelligence
AirO	air officer
Arty	artillery
ASTi	Advanced Simulation Technology inc.
Bn	battalion
C2	command and control
CAS	close air support
CFF	call for fire
Co	company
COC	Combat Operations Center
COF	conduct of fire (radio network)
DAGR	Defense Advanced GPS Receiver
DIS	Distributed Interactive Simulation
DMAO	DSEEP Multi-Architecture Overlay
DSEEP	Distributed Simulation Engineering and Execution Process
FAC	Forward Air Controller
FDC	Fire Direction Center
FiST	Fire Support Team
FO	Forward Observer
FOM	Federation Object Model
FSC	Fire Support Coordinator
FSCC	Fire Support Coordination Center
FW	fixed wing (aircraft)
GPS	Global Positioning System
IEEE	Institute of Electrical and Electronic Engineers
IR	infrared

ITACS	Information Technology and Communications Services
ITX	Integrated Training Exercise
HLA	High Level Architecture
JFO	Joint Fires Observer
JSAF	Joint Semi-Automated Forces
JTAC	Joint Terminal Attack Controller
LnO	Liaison Officer
LVCTE	Live Virtual Constructive Training Environment
LVCTT	Live Virtual Constructive Training Team
MACE	Modern Air Combat Environment
MATSS	Marine Air Training Systems Site
MCAGCC	Marine Corps Air Ground Combat Center (Twentynine Palms, CA)
MCTP	Marine Corps Training Publication
MEF	Marine Expeditionary Force
mm	millimeter
MOVES	Modeling, Virtual Environments, and Simulation
NECC	Networked Exercise Control Center
OASIS	Operationally Adaptive Simulation Integration Space
PDU	Protocol Data Unit
RDC	Remote Desktop Connection
RIU	radio interface unit
RPR	Real-time Platform Reference
RTI	Runtime Infrastructure
RW	rotary wing (aircraft)
SAVT	Supporting Arms Virtual Trainer
SDEM	Simulation Data Exchange Model
SISO	Simulation Interoperability Standards Organization
TAC	Tactical (ground maneuver radio network)
TACP	Tactical Air Control Party
TAD	Tactical Air Direction (radio network)
TEn	Tactical Environment (simulation system)
TENA	Test and Training Enabling Architecture

UDP	User Datagram Protocol
US	United States
VBS	Virtual Battle Space
VOIP	Voice Over Internet Protocol
VPN	Virtual Private Network

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

First, I must acknowledge the significant effort and contributions of my advisors, without which the Marine Corps would have one less MOVES graduate this year. Dr. Curt Blais's depth of knowledge and experience with simulation standards is surpassed only by his skill as a proofreader and writing coach. Chris Fitzpatrick is a driving force for turning broad ideas into effective and valuable theses, especially this one. Kirk Stork's technical expertise, illuminating questions, and recommendations were greatly appreciated. Many thanks for the continued guidance and support.

Second, a big thank you to the simulation operators who volunteered their time to support this project. Matt "Barf" Gurrister, in particular, spent many hours troubleshooting multiple systems to enable close air support in the simulation environment. Chris Cannon's artillery expertise and Katy "Kackle" White's aviation and close air support advice were invaluable.

The simulation training experts at MCAS Camp Pendleton went out of their way to support background research and provide recommendations for simulation environment development. A big thanks to Francis "The Chin" Pascucci of the MATSS, and Chris Getchell, Steven "Slade" Mount, and Sean Sullivan of the I MEF LVC-TE team for their hospitality, insight, and recommendations. OASIS is an impressive training capability, and a good reference point for any future interoperability efforts.

It would be a mistake to omit my companions in the MOVES cohort who were critical to my success and perseverance through two dark years. Kyle, Chris, Chris, Stefan, Bill, Bernd, and FJ: Thanks for everything. I'd also like to thank the band for their professionalism and putting on a great show.

Last, but not least, thanks are both essential and insufficient for my wife's continued support and assistance.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

Fire support coordination is a complex function that is critical to the United States Marine Corps' combined arms warfighting philosophy. It involves the integration of aviation and surface indirect fires, often in support of a ground scheme of maneuver. Training opportunities can be limited by the availability of aircraft, suitable ranges, expensive ammunition, and safety risks. The cost, risk, and limited availability of live training assets make fire support coordination training an attractive target for simulation training. Fire support coordination involves a combination of highly technical tasks, such as delivering munitions on target while flying an aircraft, and detailed coordination between distant units. Current training systems are generally focused on either developing technical skills or coordination and information management. There are no systemic training opportunities for fire support coordination participants to simultaneously practice both technical tasks and communication with other agencies.

B. PROBLEM FRAMING

The United States Marine Corps has invested heavily in high-fidelity, task-oriented training simulations, such as the flight simulators which are mandatory for portions of pilot training progression. For ground observers, the Supporting Arms Virtual Trainer (SAVT) provides a high-fidelity training environment for coordinating and integrating close air support and surface fires from computer-controlled forces. Although each of these simulation platforms support some form of technical interoperability, there is no program of record training opportunity for ground observers to practice target correlation with a pilot in a simulator. The inability of Marine Corps trainers to integrate simulation systems degrades trainee progress in preparation for rare and short-duration live training events. The intent of this project is to develop a simulation environment supporting both part-task training fidelity and command and control / coordination using available unclassified software and networks to conduct distributed fire support coordination training. The goal of this research is to explore the problem space in federating available legacy systems to demonstrate a proof-of-concept fire support coordination exercise in a simulation environment. This thesis is not a training effectiveness study; evaluation of the simulation environment consists of the research team's subjective analysis.

C. APPROACH AND ORGANIZATION

This thesis uses the 7-step Distributed Simulation Engineering and Execution Process (DSEEP) to guide the planning, development, and execution of the simulation environment (IEEE Computer Society 2010a). It is organized into the following chapters.

Chapter II—Background. This chapter discusses the current operational and training organization, systems, and roles involved in fire support coordination training. It provides an overview of the Marine Corps' current program of record training systems and efforts towards interoperability. Finally, this chapter discusses distributed simulation interoperability standard frameworks before providing an overview of the DSEEP.

Chapter III—Approach. This chapter is organized to document planning the simulation environment during steps 1–3 of the DSEEP. Step 1—Define Simulation Environment Objectives covers initial planning, resources, and expectations for the thesis. Step 2—Perform Conceptual Analysis involves the scenario design and more granularity of the desired simulation environment. Step 3—Design Simulation Environment involves detailed planning for the simulation systems and the integrated simulation environment.

Chapter IV—Implementation. This chapter covers development, integration, and testing of the simulation environment in steps 4–5 of the DSEEP. Step 4—Develop Simulation Environment includes implementing the scenario into each simulation system and confirming the network and infrastructure support simulation environment requirements. Step 5—Integrate and Test Simulation Environment covers the systemic integration of each simulation and testing to confirm desired functionality. This chapter involves multiple changes to the simulation environment as problems are identified and solutions are developed and implemented.

Chapter V—Results. This chapter documents execution and analysis of the simulation environment, guided by steps 6–7 of the DSEEP. Step 6—Execute Simulation involves a full demonstration of training scenario execution in the simulation environment. Step 7—Analyze Data and Evaluate Results includes the research team's assessment of member application suitability, simulation environment interoperability, and overall effectiveness of the simulation environment for defined training objectives.

Chapter VI—Conclusions. This chapter provides a summary of results, along with recommendations for applying this simulation environment for training and recommendations for future research efforts.

This thesis explores the technical problem space in federating available legacy systems to achieve interoperability for distributed training.

THIS PAGE INTENTIONALLY LEFT BLANK

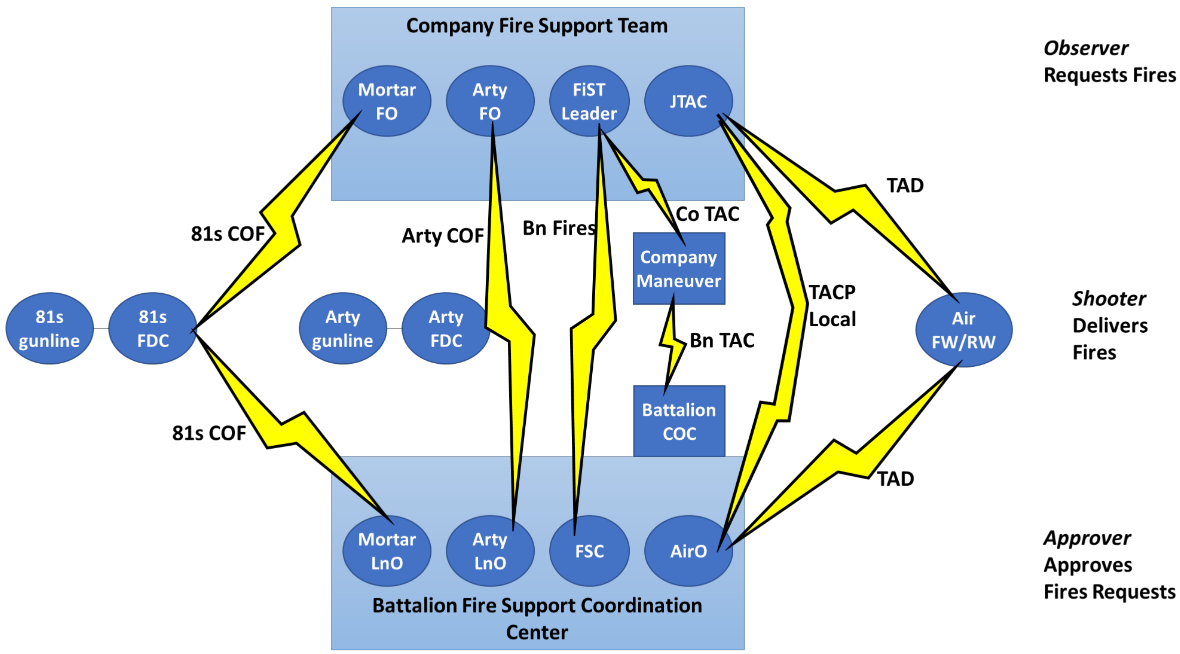
II. BACKGROUND

A. SUPPORTING ARMS ROLES IN THE GROUND COMBAT ELEMENT

Combined arms and employment of supporting fires in conjunction with maneuver is a critical component of the Marine Corps warfighting philosophy (US Marine Corps 2018b). The three essential roles in Marine Corps fires are the *observer* who requests the fires, the *approver* who has authority and awareness to authorize or deny the fires, and the *shooter* or firing agency which delivers the fires, such as artillery or aircraft. In some cases, all three roles are filled by one entity (deep air support, 60mm mortar in direct lay); in other cases, three separate entities are distributed and connected only by voice radio. This thesis will focus on these roles as they exist in an infantry or light armored reconnaissance battalion. Figure 1 illustrates a standard organization of fire support relationships. The company fire support team (FiST) serves as the observer on the top row. Firing agencies are in the middle row, including 81mm mortars, artillery, and fixed or rotary wing aircraft. The battalion fire support coordination center is on the bottom, with approval authority. The lightning bolts connecting agencies or individuals represent communication channels in the form of labeled voice radio networks.

For observers, each infantry or light armored reconnaissance company forms a fire support team (FiST) to request and control fires in support of the company's operations. The team is composed of the FiST Leader, mortar and artillery forward observers, and a joint terminal attack controller (JTAC). The mortar forward observer (FO), often a noncommissioned officer trained as a mortarman, will communicate directly with company and battalion mortar assets. The artillery FO, usually an artillery lieutenant attached from a local artillery unit, will coordinate and plan the use of artillery fires.

To request, coordinate, and control air, the FiST will be assigned either a JTAC, usually a highly qualified enlisted observer from the artillery community, or a FAC, a Marine pilot who has completed JTAC training and qualification. For simplicity, the term JTAC will be used in reference to either the FAC or JTAC assigned. In some cases, a qualified JTAC is not available, and a joint fires observer will provide observation and coordination with the aircraft on behalf of a distant JTAC who can still control the aircraft and clear strikes.



Legend:FO: Forward Observer; Arty: Artillery; FiST: Fire Support Team; JTAC: Joint Terminal Attack Controller; COF: Conduct of Fires; TAC: Tactical Maneuver; TAD: Terminal Aircraft Direction; FDC: Fire Direction Center; TACP: Tactical Air Control Party; FW: Fixed Wing (jet); RW: Rotary Wing (helicopter); LnO: Liaison Officer; FSC: Fire Support Coordinator; AirO: Air Officer

Figure 1. Fire Support Coordination Structure

The FiST leader is usually an infantry officer, often the company executive officer or weapons platoon commander, who is responsible for the integration and synthesis of all available fires to support the company's scheme of maneuver. The FiST leader must understand the capabilities and limitations of each firing agency, have close working relationships and good communication with each of his team members, and be completely integrated with the company's maneuver. The FiST leader must communicate with company and battalion leadership as well as maneuver elements, while monitoring and supervising the observers to ensure the fire support provided is integrated, synchronized, and effective. To provide responsive integrated fires, the FiST leader will decide on a broad game plan—such as "suppress anti-air defense with artillery, mark tanks with mortar illumination rounds, and destroy tanks with rotary-wing aircraft using hellfire missiles"—and ensure deconfliction and safety—"aircraft stay west of the highway to avoid artillery gun-target-line." The individual observers will then develop the details necessary for their firing agencies to confirm they can support the plan and timing and communicate their requests.

Although approval authority for fires can be retained at higher levels or delegated based on a number of factors, most supporting arms are often approved at the battalion level. The agency that approves this is the fire support coordination center (FSCC), which is always either co-located or in constant close communication with the battalion's combat operations center (COC).

> The battalion FSCC performs fire support coordination in terms of closely integrating multiple supporting arms with maneuver. It monitors and receives all fire support requests originating within the battalion. The battalion FSCC ensures that supporting arms are integrated with the scheme of maneuver and that friendly forces are not endangered. It may also coordinate missions for observers to attack targets outside the battalion's zone of action. (US Marine Corps 2018a, 1–5)

The fire support coordinator (FSC) is the individual who has the authority and responsibility to develop, execute, and supervise the battalion's fire support plan. The FSC has authority to approve or deny fires requests and must maintain full awareness of friendly unit locations and activity, firing agency location and status, and reported enemy location and activity. The battalion air officer (AirO) is a FAC who is assigned to the FSCC, and primarily works on requesting air support from higher headquarters and allocating and providing available aircraft to the company JTACs. Much like the FiST's FOs, the battalion mortar platoon and an artillery unit within the division will provide liaisons to support the FSCC. These liaisons will monitor their respective conduct of fire (COF) radio networks to record all call for fire (CFF) requests to provide the FSC the information required for decision, and then inform the firing agencies if the requests are approved, modified, or denied. Once the FSCC approves a fire mission, the firing agency will fire it as scheduled.

The FSCC's efficiency in understanding the battlespace and processing requests for fires greatly improves the responsiveness of fire support to the company FiSTs. One or more assistant FSCs can be assigned to aid the FSC or approve fires during assigned shifts to support continuous operations. The FSCC and FiSTs can be augmented by shore fire control party members to include naval surface fires in fire support planning and execution. Additionally, the FSCC is supported by clerks and communications personnel to manage information and keep radio networks and other communication channels open.

B. FIRE SUPPORT COORDINATION TRAINING OVERVIEW

Due to the technical complexity of each role in the fire support paradigm, together with the time-sensitivity, uncertainty, and high-pressure involved in integrating supporting arms into maneuver on a complex battlefield, training is of critical importance. The Marine Corps' philosophy of training in general follows a "building-block" approach, as described in MCTP 8-10A, Unit Training Management Guide:

> The training events are numbered and logically arranged from the simplest to the most complex. If conducted in order, the events provide a progressive, challenging, and building-block approach to training with specifically stated time periods for re-demonstration of combat skills. (US Marine Corps 2016, 7–7)

An example of this building-block approach starts with the training of individual skills, such as formal courses for JTAC qualification or FSC training. When individuals are proficient in their tasks, they then can contribute to collective tasks, such as the JTAC advising the FiST leader on employment of specific aircraft or ordnance as the FiST develops a fire support plan. The collective training will progress as well, as the FiST develops and communicates practice fire plans for scenarios presented on a map or sand-table, building proficiency before practicing the same individual and collective tasks in an immersive simulation, then ultimately confirming their proficiency by executing in a live-fire environment. Because costs and resources required for each stage of training increase rapidly, it becomes imperative that each agency has demonstrated progressive capability through a crawl-walk-run approach before wasting resources on an exercise they are unprepared to conduct.

The costs, availability, and risk associated with aviation flight training in particular make it an attractive domain for simulation-based training. Even expensive simulation systems provide an attractive return on investment compared with the costs and availability of real-world flight training. Still, the costs and competitive availability of high-fidelity simulation systems necessitate substantial preparation for trainees to obtain commensurate training value out of the simulation time. At the "crawl" phase, pilots often use "chair flying," a mental exercise where a pilot sitting in a regular chair (with or without the use of visual aids such as posters depicting instrument panels), pretends they are in a cockpit, and rehearses portions of the flight they are preparing for with focus on specific tasks they anticipate, such as emergency procedures. Progression to the "walk" phase can involve use of a flight simulator to rehearse an upcoming flight, followed by the "run" phase of live execution.

One principle of the Marine Corps' training philosophy is to train as you fight. Per MCTP 8-10A:

> The battle is the ultimate test of training. To train as you will fight is the fundamental principle upon which all Marine Corps training is based. Therefore, all peacetime training must reflect battlefield requirements. All leaders are considered trainers and coaches, and they must ensure that individual Marines and units receive realistic training that simulates war-time conditions. Marines' training should prepare them to perform their tasks and meet operational standards during the complex, stressful, and lethal situations they will encounter in war. If units and elements are to function together during combat, they should train together during peace-time exercises. The Marine Corps' philosophy is to train well in peace so that it can fight well in war. (US Marine Corps 2016, 1–2)

In our context, we want realistic combined-arms training with surface and aviation fires integrated with maneuver that simulates war-time conditions. Unfortunately, there are limited opportunities for ground combat units to conduct live-fire combined arms exercises with close air support due to the high cost and limited availability or resources (aircraft, fuel, maintenance, ordnance, suitable training areas or live-fire ranges). Therefore, we must use combat as our goal, and incrementally work backwards to develop a crawl-walk-run progression that uses the fewest resources possible to develop the lower-level skills before advancing to more costly and challenging training. This is often accomplished by each agency conducting separate early training, usually focused on task-oriented progressive individual then collective skills, before bringing all agencies together to train in coordination between agencies and integration of each aspect of fire support with maneuver. There are multiple training systems available throughout this progression, but most training methodologies short of live-fire training are either task-oriented, or coordination-oriented, and rarely support both effectively. The next sections provide an overview of current training systems and methodologies in order to identify capability gaps in task-oriented and coordination-oriented training systems and methodology.

C. SUPPORTING ARMS VIRTUAL TRAINER

The current program of record simulation system for training fires observers, including Joint Fires Observers (JFOs), JTACs, and full FiSTs, is the Supporting Arms Virtual Trainer (SAVT). According to the Marine Corps' Program Manager, Training Systems brochure:

> The trainer consists of a large, 15' high x 10' radius dome. Mutable high resolution projectors create a seam-less 240° horizontal and 60° vertical field-of-view image … In the SAVT system, three separate hand-held devices are used to emulate the functions of the binoculars, laser rangefinder, GPS [Global Positioning System] locator ([Defense Advanced GPS Receiver] DAGR), laser target designator, and laser illuminator. Each of the emulated devices have tracking sensors to reflect its orientation in the room, and where on the Primary Display it is marking a location in the 3-D simulated battlefield with three degrees-off freedom (3DOF) sensors. The host computer combines the 3DOF orientation data and student usage actions to generate IR [infra-red] pointer, Laser Range Finding and Laser Designation events that are usable internally and published as distributable HLA [High Level Architecture] objects. (Program Manager Training Systems 2016, 2)

Figures 2 and 3 illustrate the immersive display and high-fidelity handheld tools.

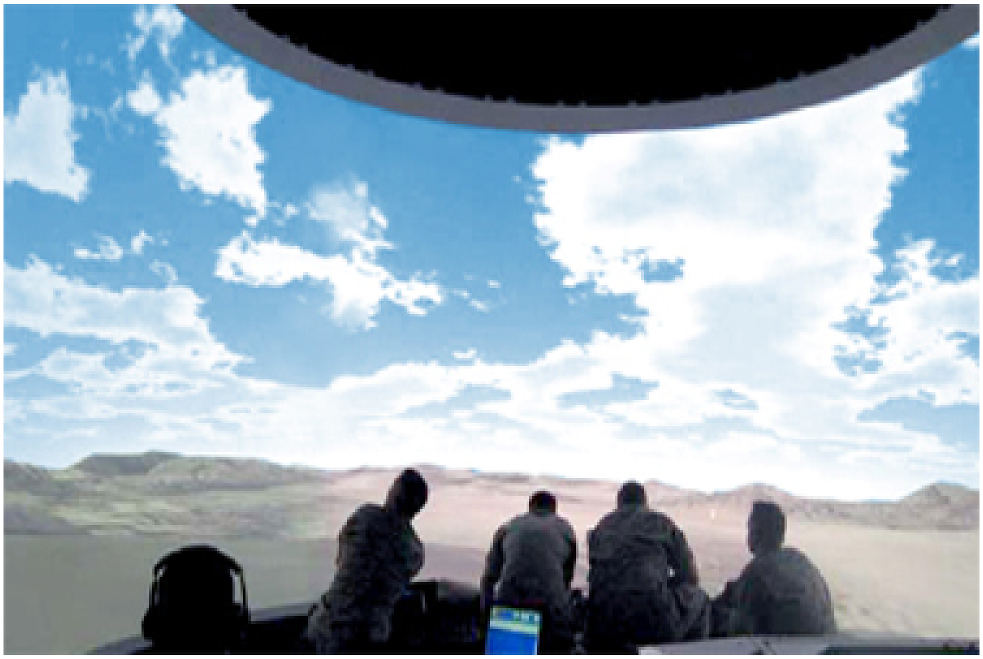


Figure 2. FiST in SAVT Simulated Environment.
Source: Program Manager Training Systems (2016).

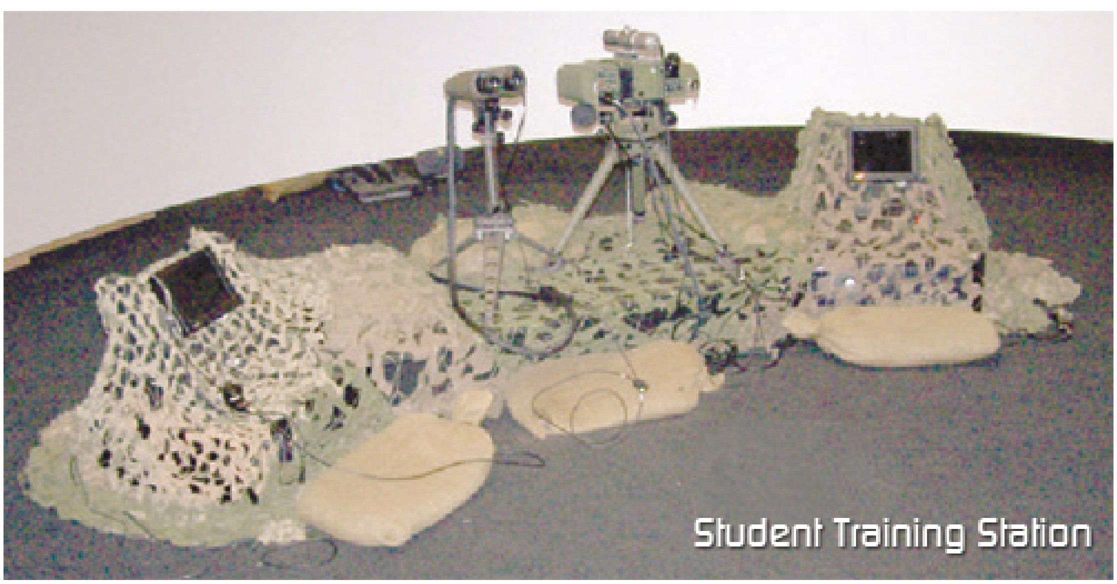


Figure 3. SAVT Instrumented Devices.
Source: Program Manager Training Systems (2016).

In a generalized training scenario, a FiST would be given the perspective of an observation post on a hill, overlooking a battlefield. A planned offensive scenario would allow the FiST to progress through a deliberate fire support plan, while changes to the scenario due to enemy action or friendly limitations would require the FiST to develop and execute quick fire plans. Depending on the complexity of training desired, friendly maneuver forces can be simulated, ensuring the FiST must integrate fires with maneuver, including linking the FiST perspective to a vehicle moving in support of the attack. The SAVT allows the FiST to plan, coordinate, execute, and visualize fires from agencies including artillery batteries, mortar platoons, fixed wing and rotary wing aircraft, and naval surface fire support.

To enable one operator to control multiple agencies, the semi-autonomous nature of the simulation is vital, but simultaneous coordination with four live trainees regarding separate assets is a significant challenge. To address this, the SAVT uses speech recognition tied to training radio interfaces to request fires and direct action from simulated agents, allowing the operator to supervise and intercede instead of personally controlling every action (Program Manager Training Systems 2016). The simulation-linked devices including optics, rangefinders, and laser designators allow the FiST to perform high-fidelity interaction with the simulation to direct fires onto target. The utility and training value are demonstrated by the fact that the "SAVT has been certified by the Joint Close Air Support Executive Steering Committee to replace 33% of live fire training controls required for JTAC annual currency training" (Program Manager Training Systems 2016, 2). The SAVT provides a useful training environment for fires observers and FiSTs to train internal procedures and individual skills in the coordination of supporting arms, both in preparation for live-fire events, and to sustain skills when those capabilities are unavailable for training. In examining the capabilities and requirements of the SAVT, Vaught (2016) estimated that a 15-minute training session using all available supporting arms would ultimately cost around the same price as a single 81mm mortar round. Clearly the SAVT is a useful and cost-effective training system, but it does have limitations.

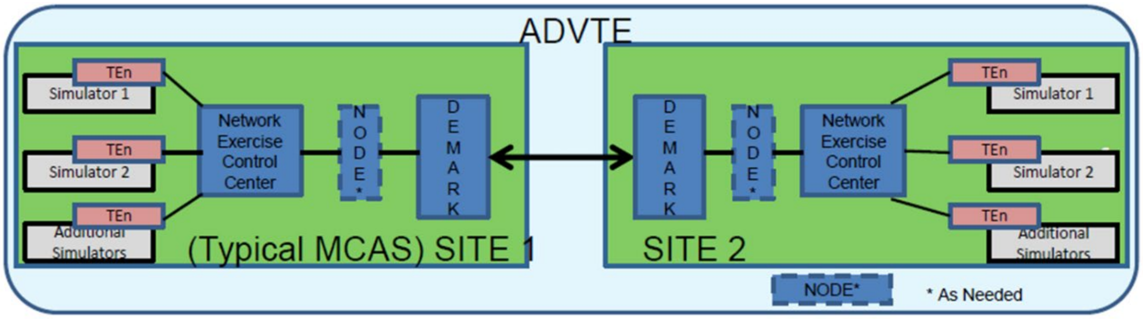
Although JSAF supports simulation interoperability using an HLA bridge, the SAVT contract specifications were for a standalone system, and does not maintain authority to operate and authority to connect with other training systems and networks. This means that although the FiST can practice internal coordination and individual skills very well, they cannot coordinate with real firing agencies, aircraft pilots, or FSCCs for training. The speech recognition software and direct interaction with the contracted operator allow the FiST to plan and execute, but are not the same as speaking with the fire direction center or a real pilot, where repeated coordination generates common understanding enabling speed and accuracy in decisions and execution. One way to integrate some human coordination is to have the Fire Support Coordinator or a close air support (CAS) pilot stand in the back of the SAVT, and conduct verbal communications with the FiST as mock radio transmissions. This can generate confidence and understanding between the FiST leader and FSC as a walk-speed building block, but because they have the same visual and auditory inputs as the FiST, it does not force the FSC to rely solely on the radio transmissions for approval. It does not generate the push and pull of information required to generate a common understanding of the operational picture, and thus does reach true training potential.

D. OPERATIONALLY ADAPTIVE SIMULATION INTEGRATION SPACE

The Marine Corps currently uses flight simulators that feature high-fidelity, functional clones of physical cockpits inside a container with a large immersive screen, such as the 270 degree horizontal by 80 degree wraparound display in the UH-1 and AH-1 simulators (FlightSafety International 2018). This allows the pilot and co-pilot to be physically present and collaborate in a realistic cockpit replica, and experience the same virtual inputs. Many of these simulators feature hydraulic systems that can provide motion with 6 degrees of freedom (movement and rotation), allowing pilots to experience motion cueing and physical sensations aligned with visual changes. Figure 4 displays the exterior of full motion flight simulators on hydraulic supports.

Figure 4. USMC AH-1Z and UH-1Y Flight Simulators. Source: FlightSafety International (2018).

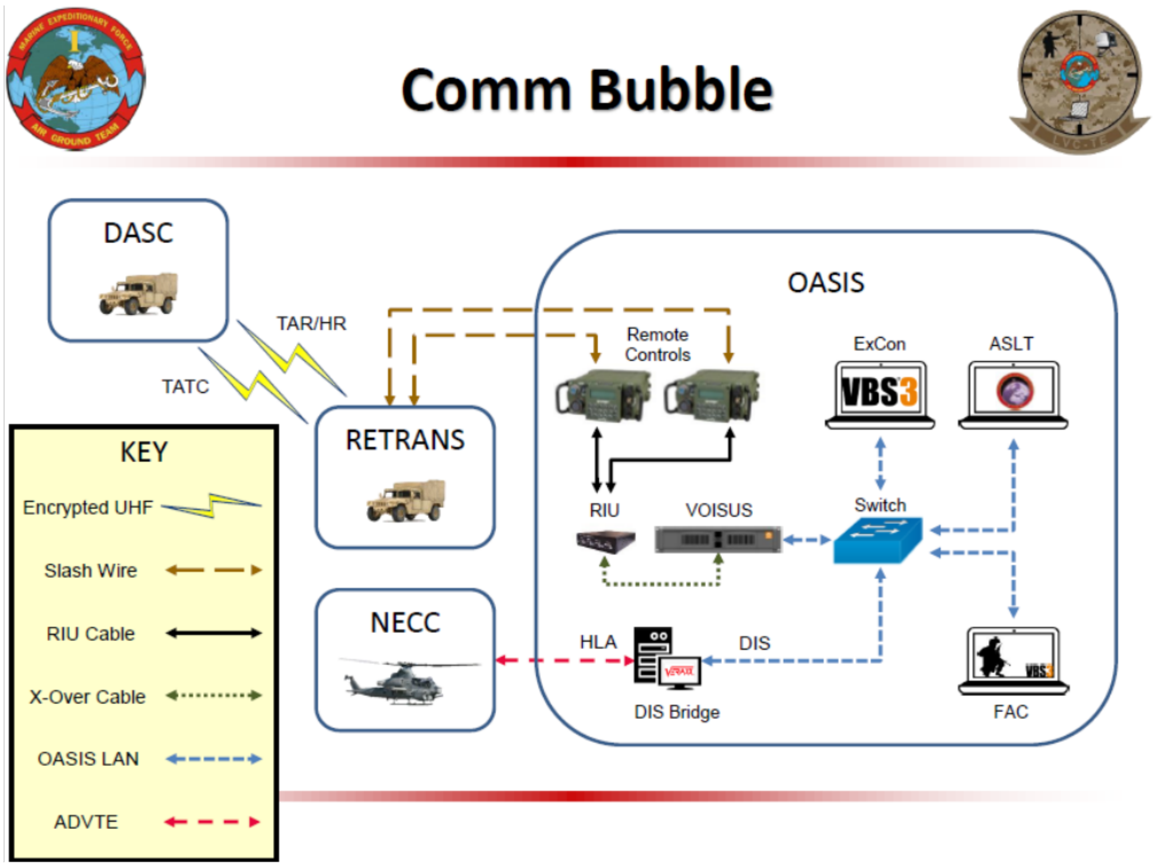
These simulators are operated at the Marine Air Training System Sites (MATSS), owned by each Marine Air Wing and located to provide training support to flying squadrons. Per the Marine Aviation Simulator Master Plan, each MATSS in the continental United States is required to have networked simulators, and each simulator must feature a Tactical Environment (TEn) which standardizes threat information, emitters, emissions, and weapons-fly-outs, and ensures USMC and joint air/ground interoperability (US Marine Corps 2017). The TEn provides each simulator a High Level Architecture (HLA) compliant networking capability with a local Network Exercise Control Center (NECC) in a basic server-client configuration. The local NECC can then be connected to distant sites via the closed-loop, encrypted, persistent Aviation Distributed Virtual Training Environment (ADVTE). Figure 5 displays two sites connecting simulators via TEn to local NECCs, which are then linked via the AVDTE to conduct distributed training (US Marine Corps 2017).



Legend: ADVTE: Aviation Deployable Virtual Training Environment; MCAS: Marine Corps Air Station; TEn: Tactical Environment

Figure 5. Networked Simulation Diagram.
Source: U.S. Marine Corps (2017).

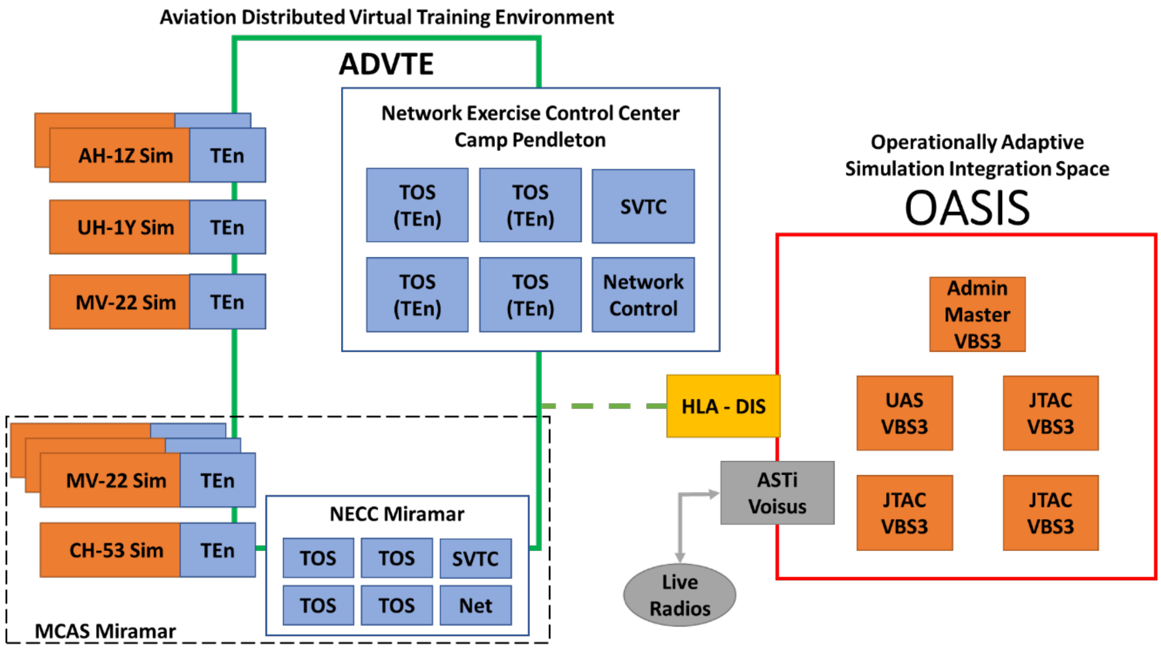
The aviation focus, classified nature of flight system capabilities, and closed-loop networks present a challenge for expanding interoperability with ground training systems. The I Marine Expeditionary Force (I MEF) Live Virtual Constructive Training Team (LVCTT) recently made progress with the Operationally Adaptive Simulation Integration Space (OASIS). Working with the 3d Marine Air Wing MATSS at Camp Pendleton, CA, they provided computers authorized to connect to classified networks and installed Bohemia Interactive Simulations' VBS3 to provide ground-oriented training simulations. Building a local network, they connected their machines and used a DIS bridge to link the DIS communications of VBS3 to the HLA communications of the NECC (Getchell 2020b). This enabled the networked flight simulators and the VBS3 machines to exchange information about friendly and enemy locations, actions, and effects. Using Advanced Simulation Technology inc. (ASTi) Voisus radio products and software, they also incorporated virtual radio communications between the systems, allowing a JTAC observing a ground target in VBS3 to communicate with a pilot in the flight simulator and coordinate close air support on a target that is commonly depicted in both systems. Additionally, they added a capability to link physical radios to the virtual radio system via a radio interface unit (RIU), allowing distant stations to communicate with air and ground simulation systems while conducting live training (Getchell 2020b). Figure 6 displays the communications linkages for a training scenario including the Direct Air Support Center to provide aviation command and control via voice radio to the pilot flying in the aviation simulator, in response to the requirements of the JTAC in the ground simulator.



Legend: ADVTE: Aviation Deployable Virtual Training Environment; ASLT: Air Support Liaison Team; DASC: Direct Air Support Center; DIS: Distributed Interactive Simulation; ExCon: Exercise Control; FAC: Forward Air Controller; HLA: High Level Architecture; LAN: local area network; NECC: Network Exercise Control Center; OASIS: Operationally Adaptive Simulation Integration Space; RIU: radio interface unit; TAR/HR: Tactical Air Request/Helicopter Request (radio net); TATC: Tactical Air Traffic Control (radio net)

Figure 6. Example OASIS Diagram. Adapted from Getchell (2020a).

Because OASIS is connected to the NECC, it can be employed with any simulator connected to ADVTE from any training site. Figure 7 displays a training exercise that the research team was able to observe, featuring multiple simulators at Marine Corps Air Station Camp Pendleton, multiple simulators at Marine Corps Air Station Miramar, and multiple JTAC positions in OASIS.



Legend: TEn: Tactical Environment; TOS: Tactical Operator Station; SVTC: Secure Video Teleconference; MCAS: Marine Corps Air Station; NECC: Network Exercise Control Center

Figure 7. Example Exercise Architecture Using ADVTE and OASIS

The exercise was technically impressive, and training participants and evaluators all provided subjective and anecdotal support and desire for the training capabilities provided by OASIS.

While OASIS makes an important step forward for interoperability of aviation and ground training simulations, it is not a comprehensive solution. The controlled environment required to limit access and provide security for classified systems at the MATSS also limits physical and networked accessibility for training. The all-purpose nature of VBS3 makes it a well-rounded platform for many training uses, but laptop screens and mouse and keyboard controls do not provide the high-fidelity experience and training value of purpose-built, task-oriented systems such as the flight simulators it connects to or the SAVT. These limitations provide an avenue forward to continue advancing simulation interoperability to provide new training capabilities.

E. LIVE VIRTUAL CONSTRUCTIVE TRAINING ENVIRONMENT

Current high-fidelity task-oriented training systems such as the SAVT or full motion flight simulators provide valuable training, but have limited interoperability. Although current flight simulators are able to connect to the Aviation Deployable Virtual Training Environment, allowing pilots to train with multiple simulated aircraft, the classified network must be partitioned from unclassified ground training systems. While small-scale local solutions such as I MEF's OASIS can connect ground forces using the medium-fidelity general-purpose simulation VBS to ADVTE along with real and simulated radios, the Marine Corps is interested in a comprehensive solution to enable interoperability between all current and future simulation systems.

In August 2020, the Marine Requirements Oversight Council approved the Capabilities Development Document for Increment 1 of a Live, Virtual, and Constructive Training Environment (LVC-TE). Although "Live, Virtual, and Constructive" simulation is a widely used term, its meaning and implementation varies, and it is valuable to understand clearly the construct and each aspect of simulation. The Department of Defense Modeling and Simulation Glossary defines live simulation as "real people operating real systems," such as training with a rifle on a firing range (Department of Defense 2011, 119). This is considered a simulation because the trainee is firing at targets or silhouettes which represent real enemies. Virtual simulation is defined as "A simulation involving real people operating simulated systems," such as a pilot operating a flight simulator (Department of Defense 2011, 159). Constructive simulations "include simulated people operating simulated systems," such as a simulated enemy surface-to-air threat deciding to fire at a simulated helicopter when it comes into range (Department of Defense 2011, 85). Although human users can provide input such as waypoints or objectives to a constructive entity, the simulation manages the performance and interactions of the entity. Constructive simulations are important for higher level planning and wargaming, allowing a division to simulate the actions of regiments and battalions without requiring a human user for every individual in the simulation. Because each of these descriptions is broad and inclusive, computer simulations often belong to multiple categories. The term "live, virtual, and constructive" is used to describe systems that incorporate any mixture of the three. With this understanding, the Marine Corp's LVC-TE intends to provide a training environment that "will enable distributed unites to efficiently and repeatedly execute combined arms training tasks, that are problematic to replicate in live environments due to range, cost, safety, and classification limitations" (Donaldson 2021, 12).

The LVC-TE implementation plan is split into two increments. Increment 1 will establish a dedicated training network and incorporate existing and legacy systems into the LVC-TE. Once that is accomplished, Increment 2 will replace modified legacy systems with a modernized training environment purpose-built for interoperability while integrating force on force and augmented reality training events (Donaldson 2021). Increment 1 will support training use cases including Company Operations, Battalion Fire Support Coordination Exercises, and Battalion Landing Team Operations Ashore. The goal is to expand the LVC-TE to support larger scale operations including MLR, MEF MSC, and MEF operations (Donaldson 2021). The systems intended to be federated or integrated into the LVC-TE during Increment 1 include the Deployable Virtual Training Environment (DVTE, which includes VBS3), the Combined Arms Command and Control Training Upgrade System (CACCTUS), the SAVT, and the ADVTE as well as command and control systems and situational awareness tools. Additional development of the LVC-TE is intended to integrate battle staff simulations, force on force systems, ground vehicle simulations, marksmanship simulations, and joint simulations (Donaldson 2021). *This thesis explores the technical problem space in federating legacy systems to achieve interoperability and distributed training.*

F. SIMULATION INTEROPERABILITY STANDARDS

To explore the challenges and opportunities involved with interoperability of networked simulations, it is necessary to understand the interoperability architectures and standards that govern them. Because the vast array of simulations utilized by the Department of Defense has a wide range of domains, purposes, requirements, and software techniques employed, it is essential that standardized protocols and architectures are defined to provide a common understanding of the virtual environment. The Department of Defense primarily utilizes three architectures or standards for interoperability between simulations for training, testing, and analysis. The Distributed Interactive Simulation (DIS) standard defines Protocol Data Units (PDUs) that each networked simulation sends to other simulations to provide the properties and activities of the entities it manages (IEEE Computer Society 2012). The High Level Architecture (HLA) incorporates standardized entity-level data and centralized Runtime Infrastructure (RTI) that provides a set of services to manage interactions between federated simulations (IEEE Computer Society 2010b). Most recently, the Test and Training Enabling Architecture (TENA) was developed to support test and training ranges with an emphasis on including live simulation data, and provides a centralized architecture as well as repositories of standard and customizable object models (Tolk 2018).

1. Distributed Interactive Simulation

The Distributed Interactive Simulation protocol grew from the Advanced Research Projects Agency's SIMNet, and was adopted by the Department of Defense in 1991 to allow users of different simulators, such as tanks and helicopters, to operate in a common virtual battlefield (Tolk 2018). Defined by IEEE Standard 1278 (IEEE Computer Society 2012), and managed by the Simulation Interoperability Standards Organization (SISO), DIS has no centralized structure and relies on each simulation providing information and updates about the entities it owns directly to the other simulations. Each simulation is then responsible for adjudicating the effects of simulation activity (such as another simulation firing a missile) upon the entities it owns. The information is exchanged between simulations in the form of Protocol Data Units (PDUs), categorized into 72 message types ranging from EntityState (includes the type, location, orientation, velocity, appearance, etc., of an entity) to Fires, Detonation, resupply actions, communications activity, simulation-specific information requests, and more (IEEE Computer Society 2012).

To ensure each simulation has a common understanding of the entities represented across various simulation domains and purposes, SISO maintains and updates the "Reference for Enumerations for Simulation Interoperability" (Simulation Interoperability Standards Organization 2020a), a library of enumerations for entities relevant to military simulation. Entities are categorized by type (platform, munition, radio, etc.), domain (land, air, etc.), country of origin, category (tank, towed artillery, communications facility, etc.), subcategory (M1 Abrams, M60 Patton, etc.), specific (M1A1, M1A2, M1A3, etc.), and extra (entity-specific usage, such as passenger capacity). These enumerations are each assigned a unique integer value that can be commonly referenced by any simulation system. For example, an M1A2 Abrams tank can be universally interpreted from 1.1.225.1.1.3, which translates to 1 (Type: Platform). 1 (Domain: Land). 225 (Country of Origin: US). 1 (Category: Tank). 1 (Subcategory: M1 Abrams). 3 (Specific: M1A2) (Simulation Interoperability Standards Organization 2020a). This exhaustive but scalable library allows each simulation to define the granularity required for its domain. A high-fidelity tank simulator will be very interested in the differences in capabilities between an M1A1 and an M1A2 Abrams tank, while a simple flight simulator not intended for close air support missions may be content to model a generic tank.

DIS simulations are often run over secure, dedicated, local networks. This is important, as the PDUs are generally sent as universal datagram protocol (UDP) packets and often multicast or broadcast (Steed and Oliveira 2010). Thus, each simulation on the network will send all PDUs "to whom it may concern" without a specific recipient identified or acknowledgment of receipt required. Each simulation, then, will also receive every PDU from every simulation on the network, and be required to interpret and assess its relevance before processing the information contained and implementing it into the simulation. This approach works well on small, closed, local networks with a limited number of simulations running, but can run into issues with more and more simulations or when distributed networks are utilized. Additionally, because each simulation is responsible for adjudication of the entities it owns, differing adjudication or disconnects in understanding between simulations can cause significant issues in representing a "fair fight." To address some of these concerns, and to improve upon capabilities incorporated in the Aggregate Level Simulation Protocol of the time, the Department of Defense established an architecture management group in 1995 to develop the High Level Architecture for department-wide standardization (Tolk 2018).

2. High Level Architecture

Defined by IEEE Standard 1516 (IEEE Computer Society 2010b), and managed by the Simulation Interoperability Standards Organization (SISO), HLA is defined by 10 rules that govern the interplay between the federation of participating simulations and systems, and the individual simulations themselves (federates). These high-level rules are broadly established to make HLA a widely applicable general-purpose interoperability architecture. Runtime Infrastructure (RTI) software provides a centralized set of services for the federation, including managing the federates as they connect and synchronize, defining the publication and subscription of information, managing the simulation objects and their ownership, managing simulation time, and optimizing the flow of data across the federation (Tolk 2018). Where DIS defines a detailed and specific protocol for simulations to share information directly with all other simulations on the network, HLA federates use a publish-and-subscribe methodology to direct information only between the systems that need it. Each federate must register with the RTI and define its subscription requirements, and during federation execution the RTI directs the flow of information to ensure each federate gets the information it requested (Knight 1998). Reducing unnecessary traffic across the network not only avoids network issues and bottlenecks, but it also reduces the workload of each simulation in receiving, interpreting, and utilizing pertinent information.

A Federation Object Model (FOM) defines the properties and characteristics of commonly understood objects and interactions. The FOM can be built from scratch or custom-modified for specific purposes in a federation, but a few widely applicable FOMs are available for use “off the shelf.” The Realtime Platform Reference FOM (RPR-FOM) provides a common representation of DIS PDUs in the HLA format, allowing legacy simulations that utilize DIS to work within an HLA Federation. RPR-FOM is commonly used and became the foundation for many follow-on activities due to its combination of HLA’s flexibility with the well-known information objects of DIS (Tolk 2018).

3. Test and Training Enabling Architecture

Because HLA is intended to be a general simulation interoperability standard, it is very broad and open, leaving room for interpretation. While this is useful for accessibility, some domains require more tightly defined structure and domain-specific standardization. The Test and Training Enabling Architecture (TENA) was purpose-built for test and training ranges, allowing it to use highly efficient domain-specific solutions such as object-oriented logical ranges, as well as common language and communications mechanisms (Tolk 2018). Built with a focus on integrating live, virtual, and constructive systems, TENA provides a common infrastructure including object model data specifications, tools, support utilities, a repository, and middleware. The middleware provides similar services to HLA's RTI, connecting range resources and analysis applications, as well as gateways for non-TENA systems to integrate. Like HLA, TENA's systems communicate with a distributed publish-and-subscribe system. With interoperability, reusability, and composability as technical requirements, TENA is a domain-specific architecture that allows new systems to integrate easily, while also allowing legacy systems that were designed as stand-alone or information stovepipes to work together in real time (Department of Defense 2016).

G. DISTRIBUTED SIMULATION ENGINEERING AND EXECUTION PROCESS

The needs, requirements, and resources across modeling and simulation communities vary significantly, and often feature applications and architectures tailored to their domain. This causes significant challenges in developing interoperability among applications, architectures, and communities. To address these challenges, SISO developed the Distributed Simulation Engineering and Execution Process (DSEEP), IEEE Standard 1730–2010 (IEEE Computer Society 2010a). This common standard is offered as a high-level framework that establishes common ground for comparison, selection, and implementation of methodologies, systems, and practices. The generalized process is intended to bridge communication barriers between users of different architectures and practices, and can be adapted and tailored to meet the requirements of a specific project and incorporate existing low-level engineering practices (IEEE Computer Society 2010a).

DSEEP identifies a sequence of seven basic steps that are required in some form to develop and execute all distributed simulation applications, whether for test and analysis or a training exercise. Figure 8 illustrates the seven steps of DSEEP, which are then briefly described.

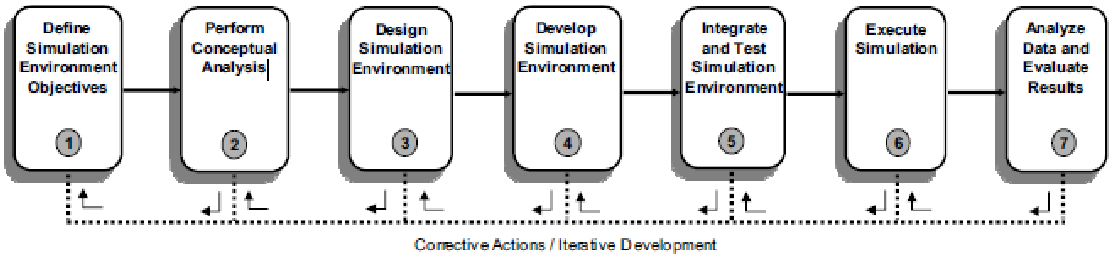


Figure 8. Distributed Simulation Engineering and Execution Process (DSEEP) Top-Level Process Flow View. Source: IEEE Computer Society 2010a.

Step 1: Define simulation environment objectives. The user, the sponsor, and the development/integration team define and agree on a set of objectives and document what must be accomplished to achieve those objectives.

Step 2: Perform conceptual analysis. The development/integration team performs scenario development and conceptual modeling, and develops the simulation environment requirements based upon the characteristics of the problem space.

Step 3: Design simulation environment. Existing member applications that are suitable for reuse are identified, design activities for member application modifications and/or new member applications are performed, required functionalities are allocated to the member application representatives, and a plan is developed for the development and implementation of the simulation environment.

Step 4: Develop simulation environment. The simulation data exchange model (SDEM) is developed, simulation environment agreements are established, and new member applications and/or modifications to existing member applications are implemented.

Step 5: Integrate and test simulation environment. Integration activities are performed, and testing is conducted to verify that interoperability requirements are being met.

Step 6: Execute simulation. The simulation is executed and the output data from the execution is preprocessed.

Step 7: Analyze data and evaluate results. The output data from the execution is analyzed and evaluated, and results are reported back to the user/sponsor. (IEEE Computer Society 2010a, 3)

Although the seven steps are sequential, many of the activities can be executed simultaneously or cyclically, and must not be interpreted as a strict lock-step progression. DSEEP breaks each step down into lower-level activities, and describes each activity along with potential inputs and outputs and a representative list of recommended tasks (IEEE Computer Society 2010a). It is not necessary to explore and describe each activity in detail at this point, but they will be included as applicable in documenting the development and execution of this thesis research. DSEEP was augmented by IEEE Standard 1730.1-2013 Recommended Practice for DSEEP Multi-Architecture Overlay (DMAO), which provides lower-level, technically focused guidance for multi-architecture (utilizing HLA and TENA, for example) development (IEEE Computer Society 2013). The DMAO also applies to issues relating to integrating simulation systems to C2 systems. There are current discussions in the SISO community about developing a C2 system-simulation system interoperation (C2SIM) overlay to the DSEEP standard. DSEEP is used to develop and execute the distributed simulation environment for this thesis research.

THIS PAGE INTENTIONALLY LEFT BLANK

III. APPROACH

Using the seven steps of DSEEP as a framework, this thesis explores the possibilities of distributed interoperable simulation training for fire support coordination. Each of the seven major development and execution steps feature lower-level activities that are recommended, but not required. Although the activities are presented sequentially for clarity, they are often conducted cyclically or concurrently, as recommended by the DSEEP (IEEE Computer Society 2010a).

A. DSEEP STEP 1. DEFINE SIMULATION ENVIRONMENT OBJECTIVES

"The purpose of Step 1 of the DSEEP is to define and document a set of needs that are to be addressed through the development and execution of a simulation environment and to transform these needs into a more detailed list of specific objectives for that environment" (IEEE Computer Society 2010a, 9). The broad objective for the simulation environment is to support a fire support coordination exercise featuring some part-task training fidelity and realistic communications architecture within a common scenario. To expand this into a detailed list of specific objectives, primary training roles and objectives are identified, exercise architecture is broadly considered, and success criteria are established.

1. Primary Training Roles and Objectives

Fire support coordination exercises can vary greatly dependent upon unit capabilities, local training range support, and availability of air and surface fires assets. The key roles in the fire support process are distilled into the following: a company Fire Support Team to observe and request fires, a close air support platform and a surface indirect fires agency to provide fires, a fire support coordination center to approve fires, a maneuver element to support, and an enemy threat. Each role has specific requirements or desired characteristics for the simulation environment, but an overarching communications architecture that reflects real world command and control is vital.

The FiST has multiple members with individual responsibilities and training objectives. The simulation environment should support some part-task training fidelity for task execution, particularly as it relates to observing the battlespace, determining friendly and target locations, and developing an integrated fires plan. The FO must be able to observe targets, call for indirect fires, adjust fires onto target, and coordinate the timing and sequencing of suppression and marks for close air support and in support of maneuver. The JTAC must be able to observe targets and friendly positions, observe aircraft and fires effects, and coordinate close air support through appropriate controls. The simulation environment should address the current training capability gap in coordinating target correlation (talk-on) with a pilot. The FiST leader must be able to observe the battlespace, coordinate with the FO and JTAC, and develop, communicate, and supervise the fire support plan in support of maneuver. The simulation environment, therefore, must provide the FiST with a ground visual perspective, identify the FiST location, and provide optics, designators, or rangefinders to identify enemy locations for prosecution. There are many first-person shooter type simulations that can provide both a detailed ground perspective as well as access to optics and far target locators.

The CAS platform must provide some part-task training fidelity for the pilot. While a multimillion-dollar full-motion flight simulator is beyond the scale and scope of this research, some level of fidelity is required to provide both physical task and coordination training. To provide a training opportunity in target correlation, a high-resolution view from the pilot's perspective is necessary. Access to sensors including day, night, and thermal optics, laser seekers, or IR designators is valuable, but will vary greatly depending upon the CAS platform and simulation system utilized. The pilot must be able to communicate with the JTAC, understand friendly positions, and identify and engage targets. A medium-fidelity flight simulator with appropriate physical flight controls as well as ground entity and weapons modeling will meet project requirements.

Although there are a few limited simulations that provide physical task training for surface indirect fires, the scope of this research prioritizes realistic coordination and delivery of fires effects. For this reason, a constructive simulation is ideal to allow a single user to fill the role of the Fire Direction Center, coordinate all fires with the observer and approver, and reduce the manpower requirement for training. A constructive simulation that can create the fires effects of an artillery battery or mortar section under control of a single user is required.

Some Fire Support Coordination Centers may have access to digital command and control systems but must be capable of approving fires using maps and whiteboards, and relying on radio inputs as their sole source of information. For this reason, the simulation environment does not need to provide the FSCC with any visualization. The FSCC must have access to the appropriate radio channels for maneuver forces, tactical air direction, and conduct of fire. If feasible, mapping simulation data to a command and control system such as C2PC could add value, but is not a baseline objective for this project.

Integrating combined arms to support a friendly scheme of maneuver is an important concept for the simulation environment. A representative friendly maneuver force must be simulated to practice the coordination of fires in support of maneuver. This maneuver force can be constructive to reduce training support requirements, or virtual to allow a maneuver unit leader to gain training value working with fire support.

Enemy forces must pose a credible threat to provide feedback on the planning and execution of fires in support of maneuver. Both air defense capabilities and a threat to ground maneuver are desired, to ensure training audiences do not fall into one-dimensional planning routines. Constructive threats are desirable to reduce the training support requirements and to ensure consistent performance and feedback within the scenario. The ability to augment threats with a virtual simulation could also allow trainees to participate as a thinking enemy, improving the variability and training value of an exercise.

Finally, the simulation environment must support a robust and realistic communications architecture. Voice radio networks must replicate the real-world communications channels between individual stations, requiring trainees to appropriately manage information flow. A voice-over-internet-protocol (VOIP) capability with separate channels to monitor and transmit on is essential. At minimum, four separate voice channels must be available for users: Fires for general fires coordination, Tactical (TAC) for maneuver, Conduct of Fire (COF) for indirect fire, and Tactical Air Direction (TAD) for aircraft. Some virtual radio software also allows for realistic communications degradation based on distance, line of sight, or atmospheric modeling, but this is not required to support this project's training objectives. The training roles and objectives are summarized in Table 1.

Table 1. Training Roles and Objectives

Role	Training Objective	Component Requirements
JTAC	Coordinate Close Air Support	Observe Targets, Friendly Forces, Aircraft Observe Fires and Effects Understand / Sense Battlespace Geometry and Timing Request Close Air Support Visual Target Correlation with Pilot Integration of CAS with Ground Fires and Maneuver
FO	Coordinate Ground Fires	Observe Targets, Friendly Forces Observe Fires and Effects Understand / Sense Battlespace Geometry and Timing Call for Fire Adjust Ground Fires onto Target Integration of Ground Indirect Fires with CAS and Maneuver
FiST Leader	Integrate Fires in Support of Maneuver	Observe Targets, Friendly Forces, Aircraft Observe Fires and Effects Understand / Sense Battlespace Geometry and Timing Coordination between Maneuver Force and Firing Agencies Integration of CAS and Ground Fires with Maneuver
CAS Pilot	Provide Close Air Support	Part-Task Training Fidelity Flight Controls Part-Task Training Fidelity Visual Perspective Observe Targets, Friendly Forces Observe Fires and Effects Understand / Sense Battlespace Geometry and Timing Coordinate Close Air Support Control Single Aircraft in Simulation Environment Employ Weapons Systems against Targets Visual Target Correlation with JTAC Conduct Close Air Support
FDC	Coordinate Indirect Fire Support	Receive and Process Calls for Fire Provide Forward Observers with Firing Data Provide Ground Indirect Fires Provide Suppression or Mark for SEAD Missions Adjust Fires per FO Requests
FSCC	Process and Approve Fires	Monitor Communications from FiST, CAS, FDC, Maneuver
Maneuver	Execute Ground Maneuver	Observe Targets, Friendly Forces, Aircraft Observe Fires and Effects Understand / Sense Battlespace Geometry and Timing Maneuver in Simulation Environment Engage Targets with Direct Fire Weapons Coordinate CAS and Ground Indirect Fires through FiST

2. Distributed Training

An objective for this simulation environment is to support distributed training. Although there are some centralized simulation centers, to make training accessible and encourage adoption, simulation systems should reside with training units, allowing for spontaneous as well as advance-coordinated training to occur. In addition, to maximize accessibility the simulation environment must be unclassified. Although this may reduce the fidelity of some weapons or flight capabilities, the accessibility for both the research team as well as potential trainees is far more important and matches the scope of the project. Ideally, the simulation environment should allow users with a decent broadband connection to conduct training from disparate sites, much like online gaming networks do. A virtual private network (VPN) connection to a purpose-built network should provide secure, reliable, and appropriately supported training from disparate physical locations.

3. Proof of Concept Scenario Execution

To demonstrate the simulation environment meets the identified goals, the research team's goal is to conduct a proof-of-concept execution of a fire support coordination exercise scenario. Representative players at separate locations will operate simulations and communicate via simulated radio against an enemy threat in a tactically realistic scenario. A FiST station will feature a JTAC and FO working together, a CAS station will feature a pilot operating a flight simulator, an artillery station will allow the FDC to control a constructive artillery battery, and an FSCC will utilize radio communications to track maneuver and approve fires. The research team anticipates abstracting and distilling individual requirements for training roles at each station to demonstrate the simulation environment's capability while reducing the manpower required to support.

In terms of DSEEP, the need identified for this simulation environment is to explore a proof-of-concept training configuration to address shortfalls identified in fire support coordination training. Because this research project is not a study of training effectiveness per se, the suitability of the simulation environment to support training requirements is based upon the subjective assessment of the research team. The standard to which the simulation environment is evaluated will be through face validation, meaning "whether a model or simulation seems reasonable to people who are knowledgeable about the system under study, based on the model's performance" (Department of Defense 2011, 101). The training members for the demonstration were selected as volunteers with appropriate training and operational backgrounds to be discussed in Chapter IV. Having defined the simulation environment's objectives, step 1 of DSEEP is complete, informing the research team's conceptual analysis.

B. DSEEP STEP 2. PERFORM CONCEPTUAL ANALYSIS

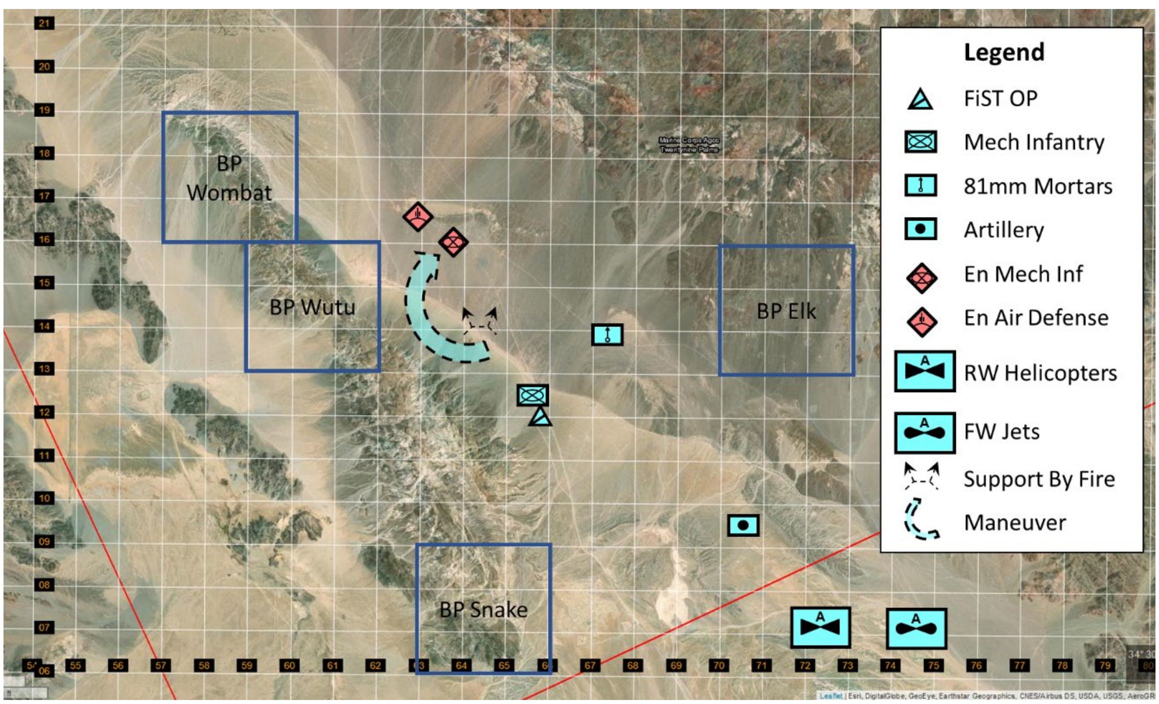
> The purpose of this step of the DSEEP is to develop an appropriate representation of the real-world domain that applies to the defined problem space and to develop the appropriate scenario. It is also in this step that the objectives for the simulation environment are transformed into a set of highly specific requirements that will be used in during [sic] design, development, testing, execution, and evaluation. (IEEE Computer Society 2010a, 13)

This step is critical to link the real-world domain described in the objectives identified in step 1 to the simulation environment that will be designed, developed, integrated, and executed in following steps. Three major activities during step 2 are to develop the scenario, develop the conceptual model, and develop simulation environment requirements.

1. Develop Scenario

To ensure the simulation environment and scenario are tactically feasible and operationally relevant, the research team referred to actual fire support coordination exercise scenarios from the Marine Corps' Integrated Training Exercise (ITX). ITX is the Marine Corps' premier service-level exercise, and is conducted multiple times a year at Marine Corps Air Ground Combat Center (MCAGCC) Twentynine Palms, CA. The vast training areas and ranges provide the best environment for Marine Corps units to conduct aviation and surface fires exercises in support of maneuver. Because ITX is designed for units from across the Marine Corps with different levels of training, experience, and proficiency in different skillsets, the exercise is built on a crawl-walk-run progression to ensure every unit is ready to train safely and successfully and gain valuable experience throughout the month-long exercise. The crawl stage consists of classroom instruction and simulator training for fires, while maneuver forces conduct small unit rehearsals and weapons drills. The walk stage is where fire support teams conduct a series of fire support coordination exercises with live indirect fire, close air support, and a small representative maneuver force. During this time, maneuver platoons and companies are conducting live-fire training without external fire support. The run stage consists of the integration of supporting air and surface fires with live-fire and maneuver, culminating in a regimental offensive operation.

The research team selected the final fire support coordination exercise of the "walk" stage, where all units involved are expected to demonstrate proficiency with the range of supporting fires before progressing to integrate with full maneuver forces. This scenario is displayed in Figure 9, and consists of a fire support team, a maneuver force with a support by fire element and an assault element both in Amphibious Assault Vehicles (AAVs), an 81mm mortar section, a 155mm artillery battery, a fixed wing section (2 jets), and a rotary wing section (2 attack/utility helicopters). The enemy threat consists of a ZSU-23-4 air defense asset and a mechanized infantry platoon with 3 BMPs and dismounted infantry in a prepared defensive position.



Legend: En: enemy; FiST OP; Fire Support Team Observation Post FW: fixed wing; Inf: infantry; Mech: mechanized; RW: rotary wing

Figure 9. Tactical Scenario

Because this tactical scenario is sufficient in complexity and realism for the final event of the fire support coordination exercises, it provides a sufficient operational context to test and demonstrate the capability and capacity of the simulation environment.

The geographical area of the scenario is the Quackenbush training area of MCAGCC Twentynine Palms. It consists of a wide desert corridor running northwest to southeast between two significant ridgelines. There is mixed scrub vegetation, broken terrain, and varying elevations interrupting sightlines throughout. An observer on the high ground will have a clear view of the entire area.

The scenario will start with all units stationary in their starting positions. As the scenario progresses, the friendly forces must suppress or destroy the enemy assets to allow the maneuver force to close with the objective and clear the enemy. The scenario ends with the enemy forces destroyed, and friendly maneuver forces consolidated on the objective.

With a tactical scenario established as a point of reference, the research team can translate real-world units and events into conceptual entities and interactions for the simulation environment.

2. Develop Conceptual Model

Entities to model in the scenario will include dismounted personnel for the FiST, AAVs for the maneuver force, 81mm mortars, 155mm howitzers, attack/utility helicopters, F18 or F35 type jets, a ZSU-23-4 air defense asset, BMPs, and dismounted troops in the defense. Additionally, the munitions used by each asset to include high explosive, smoke, and illumination rounds for artillery, and precision guided munitions from aircraft, must be consistent, along with their effects, across the simulation environment.

Key interactions will include aircraft and ground forces observing enemy positions and friendly munition impacts. Artillery suppression or destruction of enemy forces is essential to allow aircraft to close within the threat range of the ZSU-23-4, or ground maneuver to close with the BMPs. If unsuppressed, the ZSU-23-4 must be capable of targeting and shooting a RW asset in range. If unsuppressed, the BMPs must be capable of targeting and shooting AAVs within range. The AAVs should be capable of destroying suppressed BMPs or dismounted enemy in range. Aircraft fires must be capable of destroying BMPs, the ZSU-23-4, or dismounted personnel.

A generalized timeline for expected scenario execution will follow four phases. The intent is to explore and demonstrate the simulation environment's capability, vice execute a perfect tactical plan or gain training value. Phase one, preparation, will consist of communications checks, establishing the units within the scenario, acquiring targets, and initial coordination. Phase two will require coordination of air and surface fires. Artillery will suppress the ZSU-23-4, allowing close air support to destroy at least 2 of 3 BMPs. Phase three will consist of coordinating artillery suppression with maneuver. Artillery will be adjusted onto the defensive position, then duration suppression will allow the maneuver force to close with the enemy. As the assault element closes on the objective, the support by fire position will begin direct fire suppression before artillery fires cease. Once the assault force clears the objective, the maneuver force will consolidate near the objective and prepare for an enemy counterattack. Phase 4 will feature an enemy armored formation approaching the defensive position, allowing for additional close air support and artillery fires to be coordinated.

With the conceptual model established, the requirements for the simulation environment can be established.

3. Develop Simulation Environment Requirements

Having identified the goal of conducting distributed training with unclassified systems, a virtual private network (VPN) should allow users at any site with a broadband internet connection to network into the simulation environment.

To maximize the ability of existing and available simulation systems to integrate into the simulation environment, the research team plans on using DIS. As the longest running interoperability standard, it is prevalent among legacy simulation systems, and many new simulations are designed with native or add-on DIS capability to encourage integration with existing simulation environments.

The simulation environment's radio communications must support at least four channels in total, and each user must be able to monitor at least three channels simultaneously.

Live-Virtual-Constructive requirements are important to establish and clarify before designing the simulation environment. There are no live training integration requirements. Virtual simulations for the FiST and the CAS pilot are important to provide enough fidelity to support part-task training. Constructive simulations are ideal for threat systems, indirect fire agencies, ground maneuver forces, and any additional CAS assets needed. A virtual perspective for friendly or enemy ground forces could add value if available but is not critical to the success of the simulation environment.

For constructive simulations, where simulated actors take action according to their programming, it is critical to identify where and when they can operate on "auto-pilot" and where and when human in the loop control is required. Threat systems will be stationary and fire on targets they can identify in range; this requires no human in the loop. Indirect fire agencies must coordinate with observers and approvers, and fire in response to requests and feedback. This will require a human in the loop, playing the role of the FDC and translating the fires requests into simulated artillery missions. Friendly ground maneuver forces can execute a pre-planned scheme of maneuver but will need to be prompted and refined by the maneuver controller to ensure correct integration with fires. This will require human in the loop control.

Finally, the virtual environment will be represented differently in each simulation based upon the terrain databases they use. Although minor terrain discrepancies are likely without a centralized, uniform terrain database, these can be mitigated as long as there are no significant discrepancies regarding line of sight, which would reflect as one-sided cover from fires or concealment from observation. A centralized terrain database is beyond the scale and scope of this project, so deliberate testing of the objective area across simulations is important. The simulation environment requirements are captured along with the training roles and objectives in Table 2.

Table 2. Simulation Environment Requirements

Role	Training Objective	Component Requirements	Technical Requirements
JTAC	Coordinate Close Air Support	Observe Targets, Friendly Forces, Aircraft Observe Fires and Effects Understand / Sense Battlespace Geometry and Timing Request Close Air Support Visual Target Correlation with Pilot Integration of CAS with Ground Fires and Maneuver	Image Generation GPS Far Target Location Radio with CAS Aircraft Coordination with FiST Virtual
FO	Coordinate Ground Fires	Observe Targets, Friendly Forces Observe Fires and Effects Understand / Sense Battlespace Geometry and Timing Call for Fire Adjust Ground Fires onto Target Integration of Ground Indirect Fires with CAS and Maneuver	Image Generation GPS Far Target Location Radio with FDC Coordination with FiST Virtual
FiST Leader	Integrate Fires in Support of Maneuver	Observe Targets, Friendly Forces, Aircraft Observe Fires and Effects Understand / Sense Battlespace Geometry and Timing Coordination between Maneuver Force and Firing Agencies Integration of CAS and Ground Fires with Maneuver	Image Generation Radio with Maneuver Radio with FSCC Coordination with JTAC, FO Virtual
CAS Pilot	Provide Close Air Support	Part-Task Training Fidelity Flight Controls Part-Task Training Fidelity Visual Perspective Observe Targets, Friendly Forces Observe Fires and Effects Understand / Sense Battlespace Geometry and Timing Coordinate Close Air Support Control Single Aircraft in Simulation Environment Employ Weapons Systems against Targets Visual Target Correlation with JTAC Conduct Close Air Support	Image Generation Flight Controls Flight Control Weapons Control Coordination with JTAC Virtual
FDC	Coordinate Indirect Fire Support	Receive and Process Calls for Fire Provide Forward Observers with Firing Data Provide Ground Indirect Fires Provide Suppression or Mark for SEAD Missions Adjust Fires per FO Requests	Coordination with FO Fire Mission Control Message To Observer / Firing Data Constructive
FSCC	Process and Approve Fires	Monitor Communications from FiST, CAS, FDC, Maneuver	Coordination with FO, FDC, FiST, JTAC, AirO, Maneuver
Maneuver	Execute Ground Maneuver	Observe Targets, Friendly Forces, Aircraft Observe Fires and Effects Understand / Sense Battlespace Geometry and Timing Maneuver in Simulation Environment Engage Targets with Direct Fire Weapons Coordinate CAS and Ground Indirect Fires through FiST	Coordination with FiST Vehicle Control Weapons Control Virtual or Constructive
Enemy	Defend Ground Air Defense	Identify and Destroy Friendly Maneuver (if unsupressed) Identify and Destroy Friendly Aircraft (if unsupressed)	Constructive

With the conceptual analysis complete, including scenario design, development of the conceptual model, and simulation environment requirements developed, the simulation environment can be designed.

C. DSEEP STEP 3. DESIGN SIMULATION ENVIRONMENT

> The purpose of this step of the DSEEP is to produce the design of the simulation environment that will be implemented in Step 4. This involves identifying applications that will assume some defined role in the simulation environment (member applications) that are suitable for reuse, creating new member applications if required, allocating the required functionality to the

member application representatives, and developing a detailed planning documents (IEEE Computer Society 2010a, 17).

This step is key in translating the requirements and objectives into an effective plan that can be used to conduct detailed development, integration, and testing before execution. It consists of selecting the participating simulations, designing the simulation environment and interaction, designing any new applications, and preparing the detailed plan (IEEE Computer Society 2010a).

1. Select Member Applications

The member applications, or participating simulations, must be selected based on their capabilities within the roles they support as well as their integration with the scenario environment. Many factors are considered in the selection of member applications, such as cost and availability, integration with other platforms, simulation specifications and suitability, preexisting databases, or hardware requirements. Due to constraints of time and budget, simulations already available or reasonably attainable within the Naval Postgraduate School's Modeling, Virtual Environments, and Simulation (MOVES) Lab were prioritized. Sometimes the choice between two applications is not feasible at this stage, and both are considered in future steps (IEEE Computer Society 2010a). In this project, the research team attempted to identify primary and alternate candidates as applicable.

For the FiST's dismounted perspective and access to optics, designators, and rangefinders, VBS4 was selected. As a general-purpose medium-fidelity simulation system, it has organic models of ground vehicles, direct and indirect fires, aviation platforms and munitions. A built-in gateway allows control and customization of interoperability data, including DIS and HLA options. Depending on the number of computers and licenses available, members of a FiST could each have their own display and controls or share a single display. Additionally, VBS4 provides a valuable capability for virtual or constructive vehicle crews, allowing users to lead friendly or enemy ground forces in scenarios. Although the SAVT would be an ideal application for a FiST training simulation environment, cost, physical availability, and authority to connect the program of record preclude its use in this project.

To provide a CAS pilot with realistic flight performance, access to weapons systems, and native DIS capability, Lockheed Martin's PREPAR3D flight simulator was selected. A Pro Flight Trainer PUMA helicopter control device (Figure 10) provides realistic controls required for part-task training fidelity. Other flight simulation systems were considered, but the integrated weapons systems and DIS capability of PREPAR3D Professional Plus edition were important factors.

Figure 10. Pro Flight Training PUMA Helicopter Control Device. Source: Pro Flight Trainer (n.d).

To provide the constructive surface fires, Modern Air Combat Environment (MACE) was selected. Developed to support the USAF A-10 community, MACE provides constructive simulation and entity management for close air support exercises. This includes detailed modeling of ground vehicles, direct and indirect surface fires, and anti-air threats. Although other programs have some indirect fire modeling, MACE's intuitive interface is designed to allow an operator with limited fires information the ability to process and input information, provide responsive fires, and convincingly play the part of an FDC providing fires. Additionally, MACE has a similar close air support capability that allows a user to input a 9-line close air support request and assign constructive aircraft to execute. The intuitive interface and well-designed controls allow a single user with little fires expertise to direct multiple constructive agencies to provide requested fire support. MACE's intuitive and responsive entity controls make it an attractive candidate for exercise control of enemy forces, or other friendly constructive forces that require limited human in the loop control.

To provide VOIP radio capabilities, the ASTi Voisus platform was selected. Already utilized by the Marine Corps Air Ground Task Force Training Center at MCAGCC Twentynine Palms, and utilized with OASIS by the I MEF LVC-TE Team, ASTi Voisus provides robust, tailorable communications with intuitive interfaces. Additionally, ASTi Voisus can be integrated into simulation software including VBS, or utilized with live radios by using a radio bridge. Although those capabilities are beyond the scale and scope of this project, it provides potential capabilities for distributed training.

With member applications selected, the simulation environment can be designed for integration and synergy. Member applications are listed with associated training roles, objectives, and requirements in Table 3.

Table 3. Simulation Environment Requirements and Member Applications

Role	Training Objective	Component Requirements	Technical Requirements	Member Application
JTAC	Coordinate Close Air Support	Observe Targets, Friendly Forces, Aircraft Observe Fires and Effects Understand / Sense Battlespace Geometry and Timing Request Close Air Support Visual Target Correlation with Pilot Integration of CAS with Ground Fires and Maneuver	Image Generation GPS Far Target Location Radio with CAS Aircraft Coordination with FiST Virtual	VBS4 ASTi Voisus
FO	Coordinate Ground Fires	Observe Targets, Friendly Forces Observe Fires and Effects Understand / Sense Battlespace Geometry and Timing Call for Fire Adjust Ground Fires onto Target Integration of Ground Indirect Fires with CAS and Maneuver	Image Generation GPS Far Target Location Radio with FDC Coordination with FiST Virtual	VBS4 ASTi Voisus
FiST Leader	Integrate Fires in Support of Maneuver	Observe Targets, Friendly Forces, Aircraft Observe Fires and Effects Understand / Sense Battlespace Geometry and Timing Coordination between Maneuver Force and Firing Agencies Integration of CAS and Ground Fires with Maneuver	Image Generation Radio with Maneuver Radio with FSCC Coordination with JTAC, FO Virtual	VBS4 ASTi Voisus
CAS Pilot	Provide Close Air Support	Part-Task Training Fidelity Flight Controls Part-Task Training Fidelity Visual Perspective Observe Targets, Friendly Forces Observe Fires and Effects Understand / Sense Battlespace Geometry and Timing Coordinate Close Air Support Control Single Aircraft in Simulation Environment Employ Weapons Systems against Targets Visual Target Correlation with JTAC Conduct Close Air Support	Image Generation Flight Controls Flight Control Weapons Control Coordination with JTAC Virtual	PREPAR3D PFT PUMA ASTi Voisus
FDC	Coordinate Indirect Fire Support	Receive and Process Calls for Fire Provide Forward Observers with Firing Data Provide Ground Indirect Fires Provide Suppression or Mark for SEAD Missions Adjust Fires per FO Requests	Coordination with FO Fire Mission Control Message To Observer / Firing Data Constructive	MACE ASTi Voisus
FSCC	Process and Approve Fires	Monitor Communications from FiST, CAS, FDC, Maneuver	Coordination with FO, FDC, FiST, JTAC, AirO, Maneuver	ASTi Voisus
Maneuver	Execute Ground Maneuver	Observe Targets, Friendly Forces, Aircraft Observe Fires and Effects Understand / Sense Battlespace Geometry and Timing Maneuver in Simulation Environment Engage Targets with Direct Fire Weapons Coordinate CAS and Ground Indirect Fires through FiST	Coordination with FiST Vehicle Control Weapons Control Virtual or Constructive	MACE ASTi Voisus
Enemy	Ground Defense Air Defense	Identify and Destroy Friendly Maneuver (if unsupressed) Identify and Destroy Friendly Aircraft (if unsupressed)	Constructive	MACE

2. Design Simulation Environment

This activity involves assigning responsibility for entities and actions to member applications, selecting the simulation architecture, and identifying any support tools required (IEEE Computer Society 2010a). Having previously identified DIS as the simulation environment standard, member applications were selected for native DIS support. Table 4 displays each member application, its intended users, and the entities and actions it will be responsible for in the simulation environment.

Table 4. Member Application Users and Responsibilities

Application	**User**	**Responsibilities**
VBS4	FiST Maneuver	Dismounted FiST, Laser Designators Maneuver AAVs and Munitions
PREPAR3D	RW CAS	RW CAS Platform and Munitions
MACE	Arty FDC Enemy	Artillery Battery, Munitions, and Effects ZSU, BMPs, Dismounts, Emitters and Munitions
ASTi Voisus	All	VOIP Radio Traffic

To support a distributed training exercise, the DIS applications must be connected on a physical or virtual local area network. The NPS Information Technology and Communications Services (ITACS) previously established ARGON, a VPN dedicated to supporting the MOVES Institute's lab work and education regarding networked simulations. A limited use VPN connected to a local network enclave in the MOVES Lab, ARGON is the primary option for supporting the distributed simulation environment. The research team coordinated with ITACS to request ARGON be configured to support UDP broadcast, required for DIS support. Figure 11 displays graphically the initial design for the simulation environment.

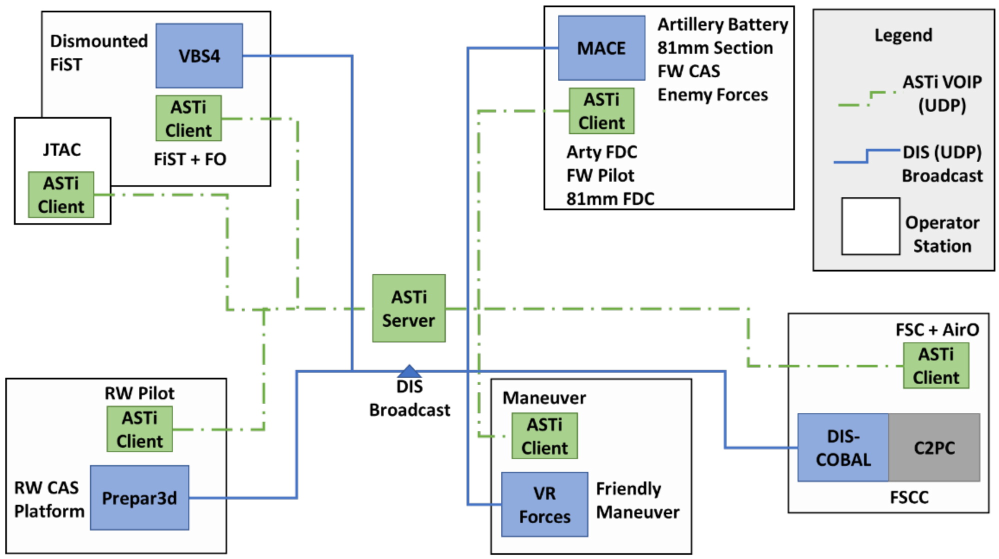


Figure 11. Initial Simulation Environment Design

Five separate operator stations connect from remote locations to the ARGON VPN and utilize UDP broadcast to transmit DIS PDUs and UDP unicast to transmit VOIP radio traffic through the ASTi Voisus Server.

3. Design Member Applications

Design, development, and integration of new member applications was unnecessary since existing simulations and supporting products were available to meet the simulation environment requirements.

4. Prepare Detailed Plan

The detailed plan is intended to provide a roadmap in the development, implementation, and execution of the simulation environment. The milestones for this project included individual configuration of each member application, local integration to ensure interoperability, establishment of VOIP radio services, establishment of the distributed network, distributed integration and testing, and finally an execution of the complete proof of concept scenario. Steps 4 through 6 of the DSEEP provide the framework for this process. Chapter IV details the development, integration, testing, and execution of the simulation environment.

IV. IMPLEMENTATION

With the simulation environment designed, the following DSEEP steps walk through the implementation of the simulation environment to best support the identified objectives. The development, integration, and testing of both the network environment, member applications, and available hardware and software resources can generate modifications and adjustments to the designed simulation environment. This chapter covers the research efforts and adjustments made to reach the desired objectives. Although this is presented in the sequential execution of the DSEEP steps, the general chronological flow was network setup, member application implementation, local integration, and remote integration. Network setup included adjusting the local network within the MOVES Lab, and working with ITACS to ensure the ARGON VPN provided the support required. Member application implementation involved the licensing and installation of member applications, as well as instantiation of scenario objects and elements within each controlling application. Once each member application had the scenario initialized, integration and testing on a local network within the MOVES lab allowed for efficient troubleshooting and collaboration. Once the research team confirmed the scenario operated properly on a local network then the additional friction of interoperability from remote locations via the ARGON VPN was explored. Many limitations and challenges identified during this process resulted in modifications to the initial simulation environment design, which are described throughout the chapter.

A. DSEEP STEP 4. DEVELOP SIMULATION ENVIRONMENT

> The purpose of this step of the DSEEP is to define the information that will be exchanged at runtime during the execution of the simulation environment, modify member applications if necessary, and prepare the simulation environment for integration and test (database development, security procedure implementation, etc.). (IEEE Computer Society 2010a, 24)

With the simulation environment designed, this step ensures the simulation team understands the detailed data that must be passed between applications, environment infrastructure, and the systemic and individual application implementation required to support the simulation environment. For repeatable simulation environments, such as persistent simulation centers designed to support a range of objectives, this step is minimal. For one-off events, this step can be incredibly involved and detailed.

1. Develop Simulation Data Exchange Model

There are two separate data exchanges involved in this simulation environment, but both were designed for simplicity in implementation, testing, and execution. The first data exchange is the radio traffic between operator stations. The ASTi Voisus server and client software are intended to be plug and play and be simple to establish and implement as long as UDP unicast is enabled between each client and the server. The research team simply had to develop the communications plan to ensure each ASTi client has access to the radio networks required to each of the distant stations it must talk to. Table 5 displays the nets required by the scenario, and which roles will need to communicate on each.

Table 5. Radio Networks and Roles

Net	Arty FDC	FiST	FSCC	Maneuver	RW Pilot
COF	X	X (FO)	X		
Fires		X	X		
TAC		X	X	X	
TAD		X (JTAC)	X (AirO)		X

The second data exchange is simulation data pertaining to the tactical scenario. Each member application natively uses DIS, providing baseline semantic and syntactic interoperability between tactical simulations. The DIS standard defines the protocol in which movement, position, orientation, fires, emissions, damage and more are shared. Each DIS application is responsible for its own entities, and adjudicates all movement, fires, detonation and effects, as they apply to the entities it owns. Each member application will then broadcast regular updates for each entity it owns to every other address on the network, and each member application will receive, interpret, and implement the updates from all other applications. This process is inefficient for larger numbers of simulation applications or complex network environments, but as long as each application understands and can display the enumerations used, DIS applications work together with an elegant simplicity. Because of this, the simulation data exchange model is simply ASTi VOIP using UDP unicast between the server and each client and DIS between all member applications using UDP broadcast. To ensure deconfliction between the DIS and ASTi traffic, the DIS traffic is broadcast to the default port of 3000 and the ASTi traffic uses the default ports of 31929. The initial simulation environment design is displayed previously in Figure 11.

2. Establish Simulation Environment Agreements

Simulation environments must have consistent rules in understanding the behaviors and interaction of simulation entities, supremacy of applications in adjudication, time management, initialization and synchronization points (IEEE Computer Society 2010a). This activity is also simplified by universal use of DIS, which establishes a standard for each of these activities and does not require development by the research team. Time management and initialization are limited, but simplified by DIS, which operates in real-time. While this limits the ability for universal time-scaling (fast forward, slow motion) for pertinent events, it does provide an effective and consistent environment for time management. Of course, real-time interaction is necessary in the current research since the configuration and execution requires humans-in-the-loop.

3. Implement Member Application Designs

In this step, the simulation environment team can expect to modify members to meet agreements, and create scenario instances in each member application (IEEE Computer Society 2010a). Native DIS support advertised in each of the member applications precludes any modifications required by the research team.

To create scenario instances, each member application has its own process to select and instantiate simulation entities. Using the tactical scenario developed in DSEEP step 2 and displayed in Figure 9, each member application instantiated the entities it is responsible for at their starting locations at MCAGCC Twentynine Palms, CA. Figures 12, 13, and 14 display the scenario implementation in each simulation.

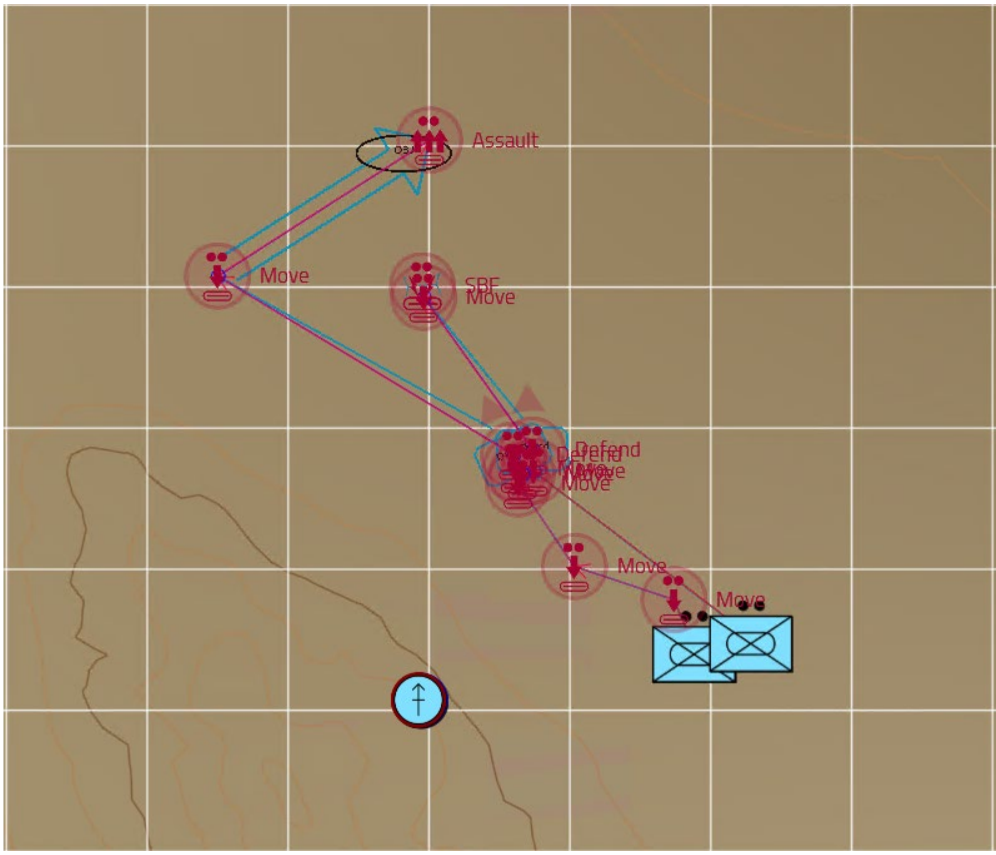


Figure 12. VBS4 Scenario Implementation with FiST and Maneuver Forces

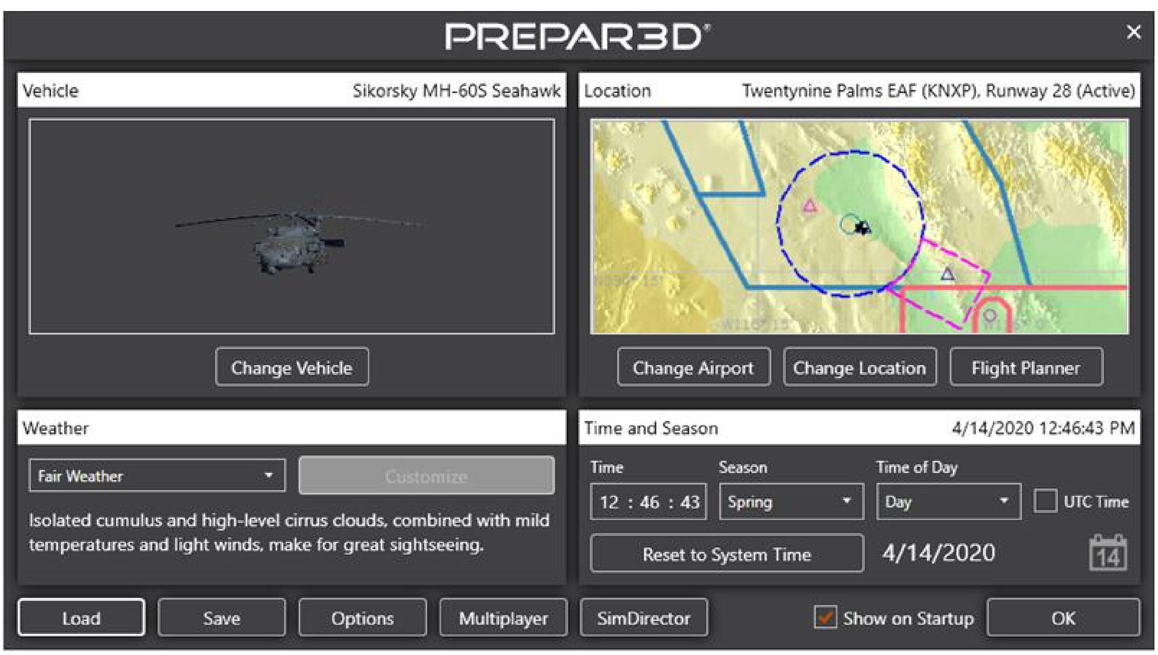


Figure 13. PREPAR3D Scenario Implementation

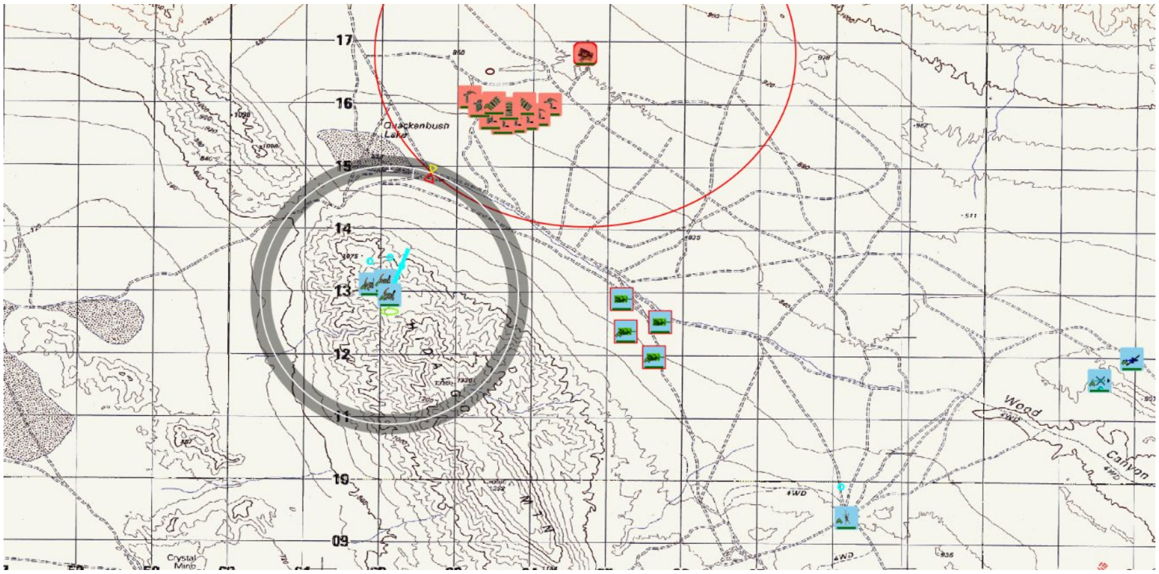
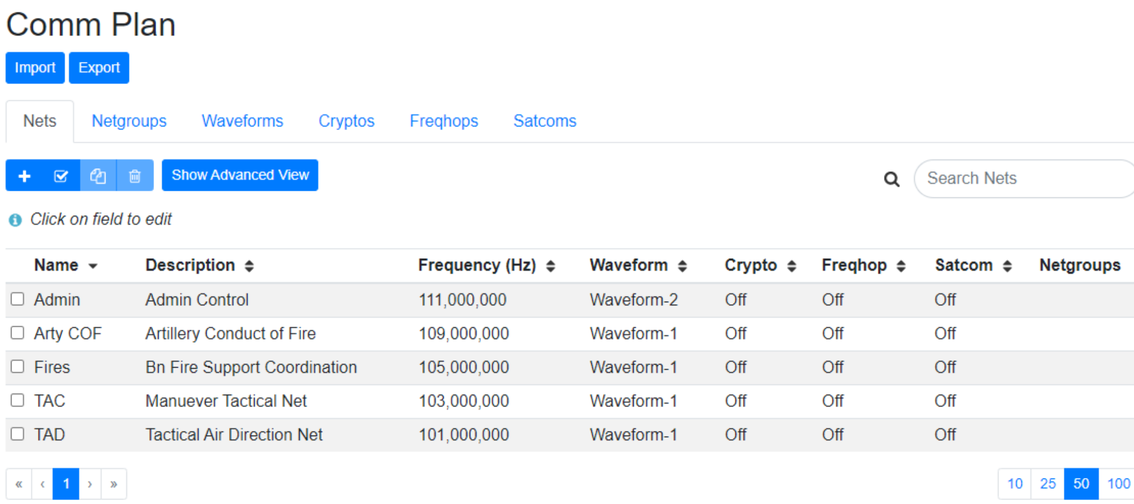


Figure 14. MACE Scenario Implementation

In VBS4, the entity representing a FAC/JTAC player with binoculars and a laser rangefinder is placed on a hilltop near the FiST OP position. In PREPAR3D, an MH-60 Blackhawk with precision guided munitions is instantiated at the airfield. In MACE, the enemy ZSU 23–4, BMPs, and dismounts are arranged on the objective with orders to defend. Friendly artillery is emplaced, and maneuver forces consisting of 4 AAVs are positioned and provided a series of waypoints that can be followed when the ground maneuver begins. Each of the DIS member applications has the ability to save the scenario, and load the scenario as required to initialize or reset the entities involved.

In addition to each of the DIS member applications, the ASTi Voisus design must be implemented. The server configuration is accessed through a browser and is reasonably intuitive and user-friendly. Configuration begins with creating the communications plan, by establishing each radio network that is available with options to change the waveform, frequency, radio equipment and more as displayed in Figure 15.



Name	Description	Frequency (Hz)	Waveform	Crypto	Freqhop	Satcom	Netgroups
Admin	Admin Control	111,000,000	Waveform-2	Off	Off	Off	
Arty COF	Artillery Conduct of Fire	109,000,000	Waveform-1	Off	Off	Off	
Fires	Bn Fire Support Coordination	105,000,000	Waveform-1	Off	Off	Off	
TAC	Manuever Tactical Net	103,000,000	Waveform-1	Off	Off	Off	
TAD	Tactical Air Direction Net	101,000,000	Waveform-1	Off	Off	Off	

Figure 15. ASTi Voisus Comm Plan

Many of these options and the fidelity of radio communications modeling exceeded the requirements of this project. Following this set-up, user roles must be defined to ensure each client that connects has access to the correct networks and assets. An example of the Roles creation is displayed for the Maneuver role in Figure 16, and Figure 17 shows the configuration of that role's second radio to access the Battalion Fire Support Coordination net.

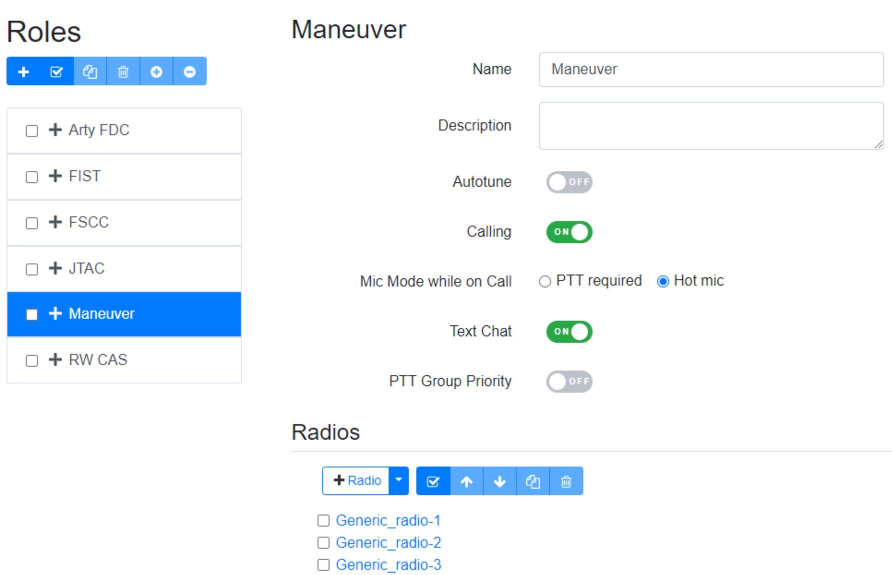


Figure 16. ASTi Voisus User Roles

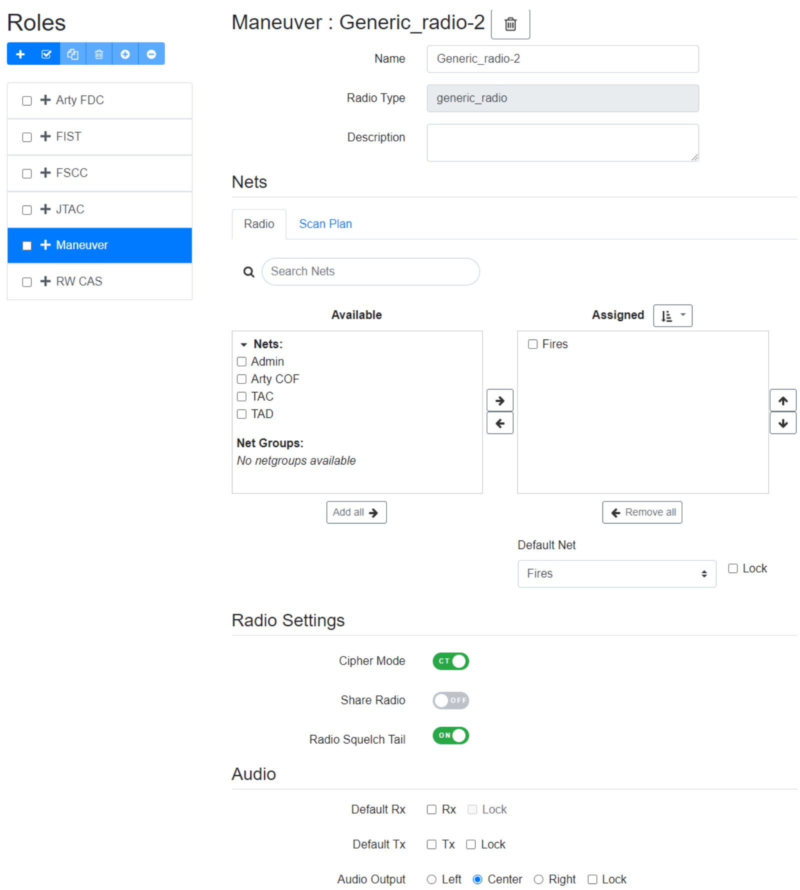
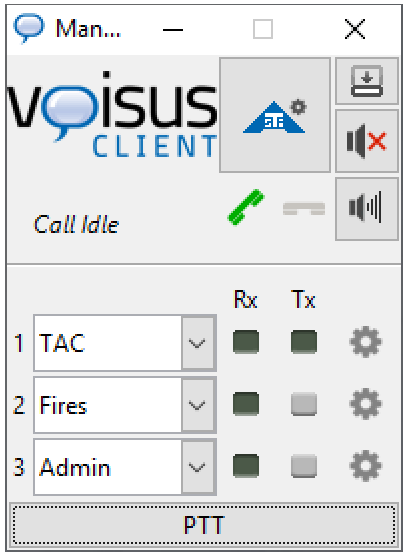


Figure 17. ASTi Voisus Radio Settings

The Comm Plan and User Roles were created to support the requirements identified in Table 5. This allows any client to connect to the local server, select their role, and have access to clearly labeled radio networks. Figure 18 displays the minimalist ASTi client interface.



Legend: TAC: Maneuver tactical net; Rx: Receive; TX: Transmit; PTT: Push-to-talk

Figure 18. ASTi Voisus Client

This simple interface allows a user to easily select which of the available networks each radio will be set to, which radios will be monitored with the receive (Rx) column, and which single radio will transmit (Tx) when the push to talk button is pressed. Additional options can adjust audio levels or assign any key to the push to talk role to allow for transmission within a full screen simulation. With the scenario implemented in each simulation application and the ASTi Voisus software, the research team is ready for the next activity.

4. Implement Simulation Environment Infrastructure

> The purpose of this activity is to implement, configure, and initialize the infrastructure necessary to support the simulation environment and verify that it can support the execution and intercommunication of all member applications. This involves the implementation of the network design [e.g., wide area networks (WANs), local area networks (LANs)]; the initialization and configuration of the network elements (e.g., routers, bridges); and the installation and configuration of supporting software on all computer systems. This also involves whatever facility preparation is necessary to support integration and test activities. (IEEE Computer Society 2010a, 28)

Simulation environments intended for temporary locations, such as field exercises or single-use modifications, must ensure everything from power and climate control to network infrastructure is in place before attempting to integrate. With the MOVES Lab designed to support numerous computers, and remote locations for distributed integration being home offices currently supporting distance learning, this was not a concern for the research team. The MOVES Lab already had network infrastructure required to support local integration and required no additional effort at this stage. Of greater concern was the configuration of the ARGON VPN to support UDP broadcast needed for DIS interoperability. Because the research team was dependent on external support from ITACS, and technical and security concerns were prevalent and uncertain, requests were made months in advance of planned integration. Due to a misunderstanding between the requirements of this research effort and other MOVES Lab requirements for education support, the research team proceeded for months under the understanding that the ARGON VPN could be configured to support UDP broadcast. In early testing of the network, remote systems could not receive any UDP packets broadcast through the VPN.

In a conference with ITACS and VPN specialists, it was discovered that it is technically impossible to use UDP broadcast or multicast through a VPN. Because UDP broadcast packets are sent to all available addresses on the network, once they are transmitted through the VPN they do not contain the routing information necessary to then be forwarded to other addresses on the network. This limitation caused the first significant change to the simulation environment design. Instead of operating DIS member applications at remote locations, they were installed on local machines in the MOVES Lab. Because each of the DIS applications are on the local network in the MOVES Lab, UDP broadcast is available and DIS interoperability is technically feasible. Operators at remote locations could then connect via VPN and utilize remote desktop connections to control the simulation applications. This allows for local interoperability between DIS simulations and ASTi VOIP clients while also supporting operation from remote locations. Figure 19 displays the updated simulation environment.

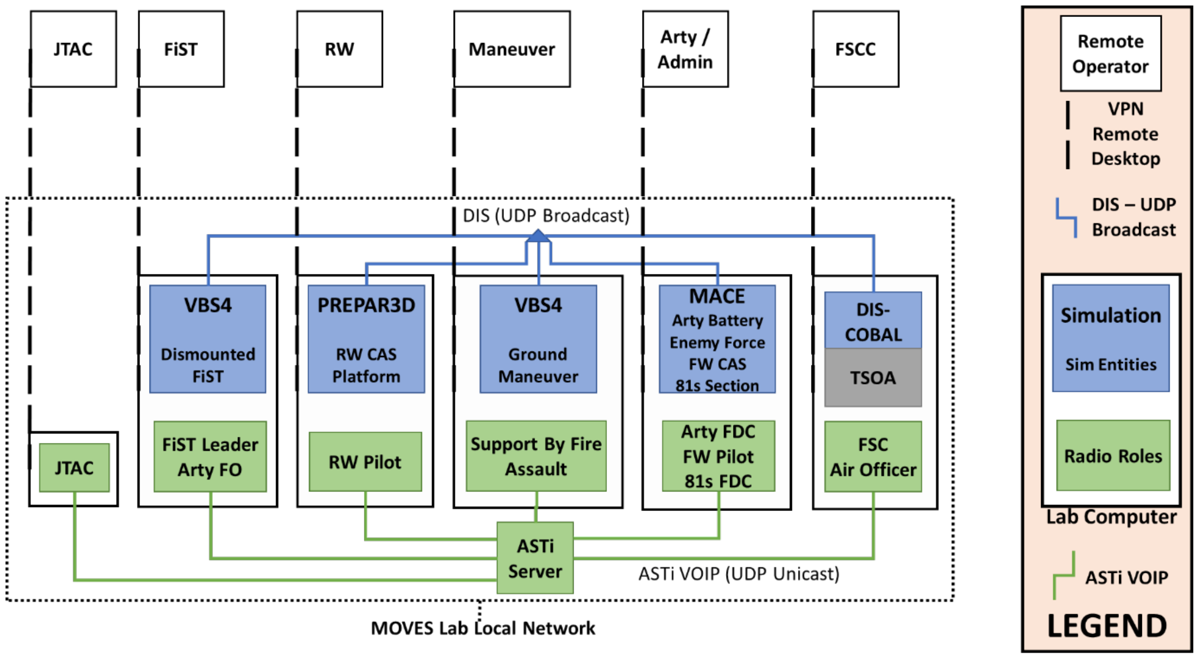


Figure 19. Updated Simulation Environment

With the ARGON VPN already demonstrating support for remote desktop control of machines in the MOVES Lab, the simulation environment infrastructure is implemented, completing DSEEP step 4 and enabling the integration and testing of the simulation environment.

B. DSEEP STEP 5. INTEGRATE AND TEST SIMULATION ENVIRONMENT

"The purpose of this step of the DSEEP is to plan the execution of the simulation, establish all required interconnectivity between member applications, and test the simulation environment prior to execution (IEEE Computer Society 2010a, 29)." The main activities identified during this step are planning execution, integrating the simulation environment, and testing the simulation environment.

1. Plan Execution

> The main purpose of this activity is to fully describe the execution environment and develop an execution plan… This planning should address which personnel will be operating the member applications (operational personnel) or supporting the simulation execution (support personnel) in other ways (e.g., monitoring, data logging). It should detail the schedule for

> both the execution runs and the necessary preparation prior to each run. (IEEE Computer Society 2010a, 29–30)

For this research, "execution" refers to the demonstration of the scenario within the environment as a proof of concept. This caused the planning and execution to be conducted differently than a simulation environment designed for a training exercise or test or evaluation. In a training exercise, training objectives determine key actions or decision points required for trainees to face. A Master Scenario Event List (MSEL) is often generated to ensure a consistent and believable scenario leads to desired decision points or task execution. Because the research objective is to demonstrate and assess the capabilities of the simulation environment, the research team prescribed the broad scenario execution to ensure desired interactions and mechanics are included. In this case, the MSEL prescribes not only environmental and enemy action to prompt decisions or actions, but also pre-determines the major decisions and actions for each operator. Appendix A displays the MSEL for scenario execution. It provides a chronological list of events to occur in the scenario, including the role or actor, trigger to execute the event, and detailed simulation activity and radio traffic involved. With the scenario execution scripted in the detail required for execution, identifying personnel to support these tasks and radio calls can be done. Table 6 outlines the participants along with their stations, roles, callsigns, radio nets, and simulation system to operate.

Table 6. Scenario Operator Laydown

Station	Role	Callsign	Nets	Sim	Operator
FiST	FiST Ldr	Ridgeback 5	TAC, Fires	VBS4	1
	FO	Ridgeback FO	COF	VBS4	1
	JTAC	Ridgeback	TAD	VBS4	2
FSCC	FSC	Trident Fires	TAC, Fires, COF		3
	AirO	Trident Air	TAD		3
RW	RW	Stinger 62	TAD	Prepare3d	4
Arty	Arty	Brimstone	COF	MACE	5
En	ZSU			MACE	5
	Defense			MACE	5
	CAtk			MACE	5
Mnvr	SBF	Ridgeback 2B	TAC	VBS4/MACE	6
	Aslt	Ridgeback 2A	TAC	VBS4/MACE	6

With detailed plans prepared, the research team was able to integrate and test the simulation environment to ensure it supported the scenario execution.

2. Integrate and Test Simulation Environment

While the DSEEP provides two separate activities for integrating and testing the simulation environment, a common software development approach is to "implement a little, test a little" in a progressive spiral instead of a lock-step waterfall approach. In an Interservice/Industry Training, Simulation and Education Conference (I/ITSEC) tutorial presentation, O'Connor and LeSeur recommended seven activity blocks, or broad steps for the integration and testing of simulation environments for LVC training, test, and evaluation purposes. Much like the DSEEP, these recommended best practices are intended to be adapted, tailored, or ignored as applicable to the intended simulation environment. O'Connor and LeSeur's seven activity blocks (2020) are:

1. Network Characterization, Collaboration Tools
2. System/Simulation Interoperability
3. Scenario Thread & Data Collection
4. Full Scenario & Analysis
5. Operator Training
6. Role Player Training
7. Dry Runs

The research team did not execute the seven activity blocks in lock step but did follow the general progression and incorporated many of the concepts and guidance as they applied to this project. Given an established VPN and local network and implemented scenarios in member applications the broad chronological steps for integration and testing of the simulation environment were to conduct local integration and testing in the MOVES lab, followed by remote integration and testing through the ARGON VPN. These chronological steps are presented in O'Connor and LeSeur's activity blocks for clarity.

a. Network Characterization, Collaboration Tools

Even with an established network, O'Connor and LeSeur's first activity block, Network Characterization, Collaboration Tools, is essential. The research team started by confirming network hardware and software supported the connectivity between computer systems. This involved connecting to the local or remote network and conducting pinging between network addresses to ensure basic connectivity. Bridging the gap to O'Connor and LeSeur's second step, System/Simulation Interoperability, the research team utilized MAK technology's DIS Spy application to receive, interpret, and record the DIS PDUs on the network.

b. System/Simulation Interoperability

To confirm that the systems could send meaningful data, DIS Spy allowed the research team to identify that the ARGON VPN did not allow remote machines to broadcast DIS PDUs, and focused troubleshooting on the network instead of the simulation's implementation of (or failure to send or receive) PDUs. Figure 20 displays a DIS Spy screenshot showing an overview of DIS traffic sent on the local network from VBS4, PREPAR3D, MACE, and the ASTi Voisus server, respectively. DIS Spy provides a variety of options, displaying DIS PDU information by network address, entities, events, and other data. Once the team confirmed each simulation can produce and send DIS PDUs, and each machine can receive the PDUs from other addresses, it is important to ensure the simulation applications are correctly interpreting and implementing the entities and events broadcast from other systems.

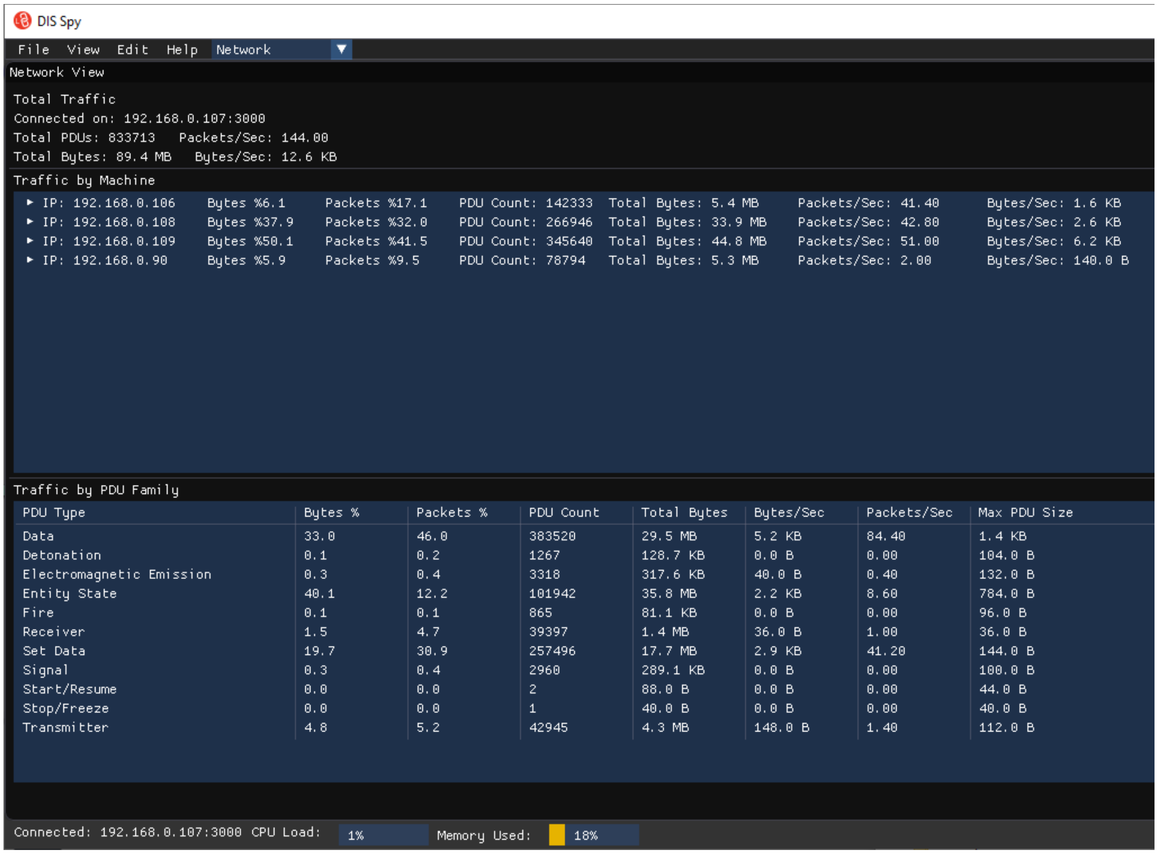


Figure 20. DIS Spy Summary of DIS Network Traffic

With each member application running an instantiation of the scenario, the research team methodically checked each entity and event in the other applications to ensure they were implemented properly. First running VBS4 and MACE, the research team worked through the dismounted FiST members, AAVs of the maneuver element, BMPs, dismounted infantry, and ZSU-23-4 of the enemy position. The armored vehicles all were interpreted and implemented correctly in both simulation applications, but dismounted infantry were not appearing properly. Figure 21 shows a screen capture of the VBS Gateway, which controls DIS implementation, as well as HLA and other interoperability protocols. It provides settings for different standards and protocols, as well as listing the details and implementation of all entities and events that are included in the simulation scenario.

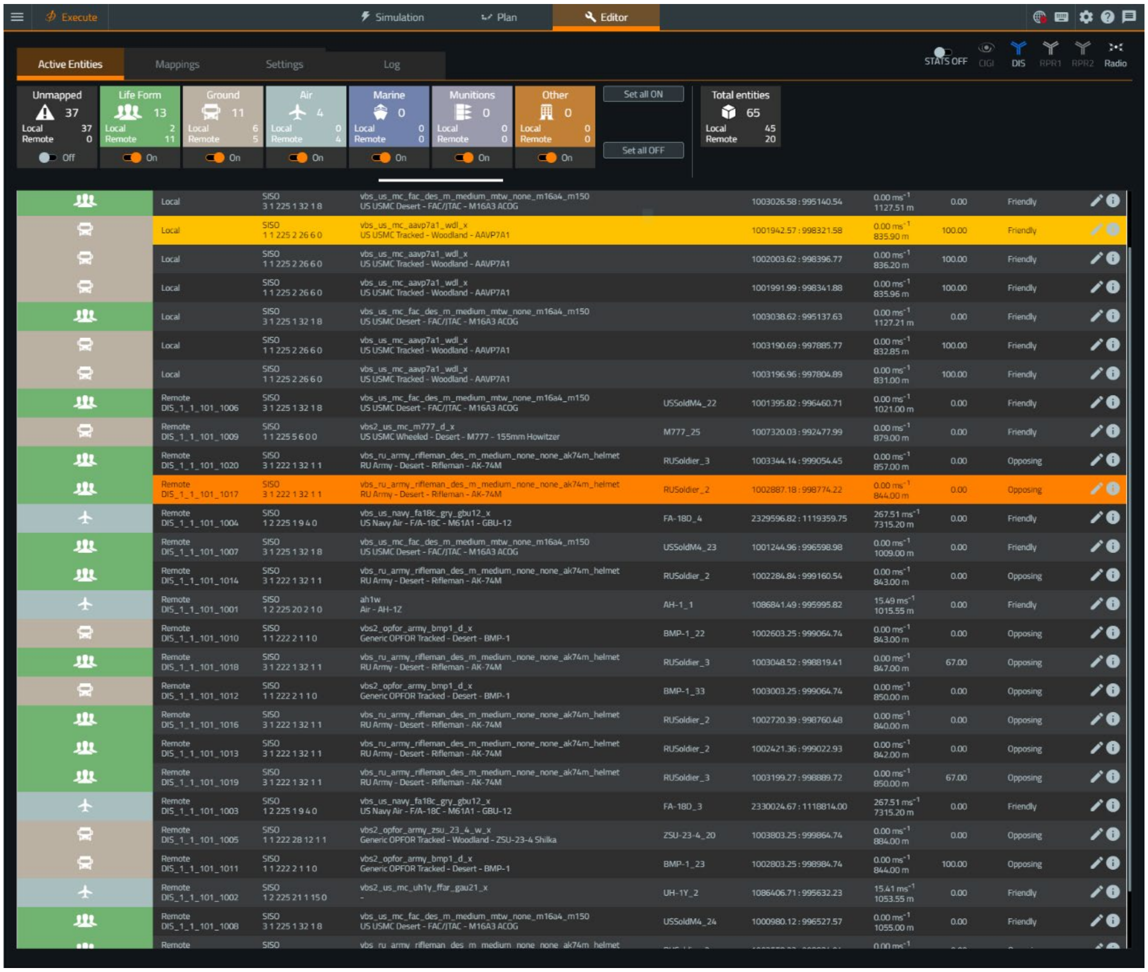


Figure 21. VBS Gateway Entity List

Using VBS Gateway, the research team found that the enemy infantry transmitted by MACE were mapped to an enumeration that was not specifically modeled in VBS4. The MACE infantry had a weapons loadout that was not exactly the same as the options for enemy infantry in VBS4. By enabling "fuzzy mapping" in the VBS4 DIS settings, VBS4 began to attempt to display and implement a "close enough" option, which was effective in displaying the dismounted OPFOR infantry. DIS Spy revealed that there was no entity transmitted from VBS4 to represent the JTAC. The VBS Gateway showed that the specific loadout of "USMC FAC/JTAC with M16/ACOG" was not mapped to a specific DIS enumeration. By manually mapping the VBS4 entity to the same enumeration that MACE used for friendly dismounted infantry, VBS4 began transmitting PDUs for that entity, and it immediately appeared correctly within MACE. Figure 22 displays an example of setting a custom mapping for entities within the VBS Gateway, specifically confirming that MACE’s enemy infantry entities are mapped to a VBS4 entity that can be rendered.

Figure 22. VBS Gateway Entity Mapping

With system interoperability confirmed between MACE and VBS4, the research team connected PREPAR3D. Once DIS was enabled in PREPAR3D’s network settings, PREPAR3D’s entities—an MH-60 helicopter and enemy BMP1 and ZSU-23-4’s targets—appeared immediately in VBS4 and MACE. Unfortunately, PREPAR3D did not display any of the entities broadcast by VBS4 and MACE, even though the PDUs were received by the system. Adjusting entity mapping in VBS4 to ensure the enumerations for VBS4 BMPs matched exactly to the enumerations provided by PREPAR3D did not resolve this issue. Exploring this issue revealed a known bug in PREPAR3D v5.1, which was pending a numbered update to resolve. With send-only DIS capabilities unacceptable for the project, the research team reverted to PREPAR3D v4, which has demonstrated full DIS support. Upon enabling DIS in PREPAR3D v4, all MACE and VBS4 entities appeared immediately with a "fuzzy mapping" interpretation. Because PREPAR3D did not implement an AAV model, it displayed the friendly maneuver element as a M113 Armored Personnel Carriers. Although not visually precise, this still allows the pilot to distinguish between friendly maneuver and enemy armored vehicles and supports the tasks required for the scenario. Once the research team confirmed that each simulation system could send, receive, and interpret the PDUs for the entities controlled by each other simulation, O'Connor and LeSeur's third activity block could be pursued.

c. Scenario Thread and Data Collection Testing

To focus integration testing, O'Connor and LeSeur distinguish between the full scenario and scenario threads. The full scenario is the totality of the simulation environment and all activities, entities, and participants, while a scenario thread is a single chain of events in a shorter duration and includes only a subset of participants (2020). By identifying critical threads which are key to the scenario, are repeated multiple times, involve very complex activities, or involve activities that have not been performed before, the simulation team can test and integrate "building blocks" which eliminate issues at a smaller scale before attempting the full scenario (O'Connor and Leseur, 2020).

The research team identified two key threads for integration testing: integration of aviation fires with surface fires in a suppression of enemy air defense (SEAD) mission, and integration of surface fires with maneuver in a closure series. In this case, the artillery battery will provide 2 minutes of suppression on the ZSU-23-4 and mark the BMP position for CAS to destroy. To test the SEAD thread, the research team ensured key interactions were implemented between simulations (visualization of targets, visualization of marking fires, visualization of suppression fires, and correct effects of aviation fires munitions on targets). On the local network in the MOVES Lab, three participants operated PREPAR3D, VBS4, and MACE while communicating on the ASTi Voisus clients. The research team quickly confirmed that both VBS4 and PREPAR3D users could visualize both marking and suppression fires executed by the constructive artillery battery in MACE. PREPAR3D is focused on modeling flight performance and control and does not provide high fidelity implementation of sensors or laser targeting devices that would be available on CAS aircraft. The pilot providing CAS in PREPAR3D was still able to visually acquire the target entities, as well as select a target for precision weapons engagement from a list. Although this is not a high-fidelity training opportunity for ground or aviation laser guided precision munitions, it does provide a precision munition capability that can still support fire support coordination and integration. The precision munitions are fire and forget, and automatically seek the selected target entity if fired in the approximate direction. Although the munitions and detonations were implemented across each simulation, there was a disconnect with how the ground vehicles were damaged by the detonations. PREPAR3D has limited modeling of damage to ground vehicles, with each entity listed as either undamaged or destroyed. This does not match the health percentage modeled in MACE and VBS4, which allows for incremental damage to build up to destruction. As a result, the pilot in PREPAR3D was shown destroyed vehicles while VBS4 and MACE showed them as damaged but still effective. To compensate for this, the MACE operator kept a close eye on ground targets and would manually set an entity to destroyed if it were damaged by aviation munitions.

Because PREPAR3D and VBS4 each have their own terrain database, the terrain displayed in each simulation was not identical. The macro terrain, representing large ridges and valleys, was close enough for big picture correlation, but micro terrain relating to vegetation and desert terrain textures appeared differently to each observer. Figures 23–26 display the differences in terrain rendering between VBS4 and PREPAR3D.

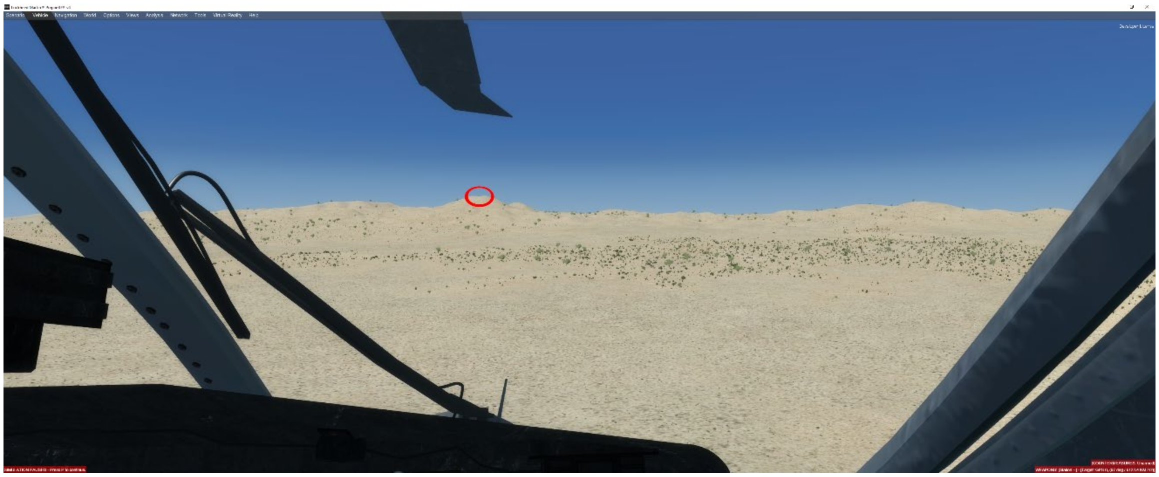


Figure 23. Macro-terrain View from PREPAR3D

Figure 24. Macro-terrain View from VBS4

The macro-terrain of the ridgeline northeast of the objective area is displayed, with the highest peak circled in red for clarity. Comparison of the ridgeline between the two simulation systems shows differing texture patterns, but the general shape of the ridge provides common landmarks for reference.

Figure 25. Micro-terrain View from PREPAR3D

Figure 26. Micro-terrain View from VBS4

The micro-terrain views in both simulations display the difference in terrain and vegetation presentation. The ZSU-23-4 is circled in red in both images. Although the helicopter is in the same position in both images, the target locations appear dramatically different. If feasible, reconciling these terrains would be an important step for increased training value in target correlation for close air support. For this research project, the time and resources available precluded detailed synchronization of terrain textures and vegetation, so an artillery smoke mark for SEAD was used as a common reference point for target correlation.

Working through the CAS and SEAD processes, a CAS time-on-target (TOT) is often designated as the point where aviation fires first strike the target. This allows observers and firing agencies to schedule fires before, simultaneous, or after the TOT. Because DIS does not support time management, there was no common scenario time between simulation systems, so the research team synchronized wristwatch time and used a common "real-world" time to coordinate timing of fires. Although MACE allows for a CAS TOT in the timing of fires, these are tied to the scenario time, which is not shared across the simulation environment. To execute this properly, the MACE operator playing the artillery FDC has to back-plan the timing from the expected impact and compensate for the time of flight of a volley, then manually execute the fire mission based on wristwatch time. Figure 27 displays the MACE artillery call for fire interface.

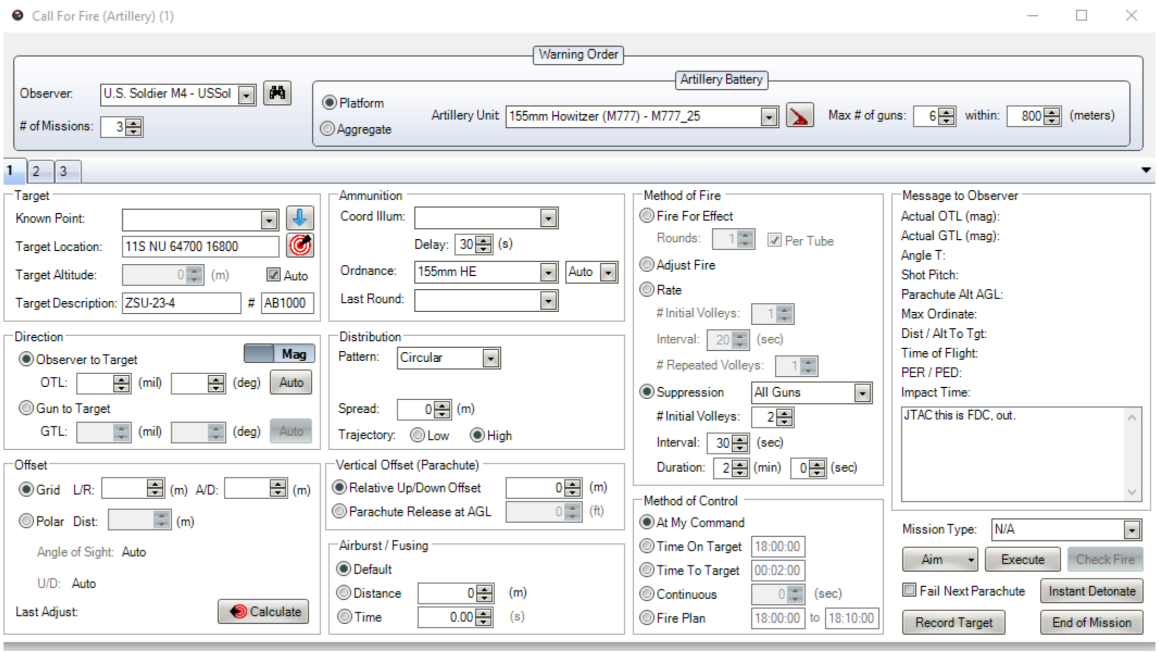


Figure 27. MACE Call for Fire Interface

With these key interactions confirmed, the research team was able to rehearse a close air support attack in conjunction with an artillery suppression of enemy air defenses, confirming the scenario thread is effective and repeatable.

The second key thread, integration of fires and maneuver, can be confirmed with a closure series. A closure series is used for maneuver elements to close on a hostile objective and echelons the available supporting fires to keep the enemy position suppressed until the assault force is on the objective. In this case, artillery suppression allows the maneuver forces to move from an attack position, establish a support by fire position, transition from artillery to direct fire suppression as the assault element moves within the minimum safe distance, and cease direct fire suppression as the assault element closes on the objective. While both VBS4 and MACE were capable of providing a constructive maneuver element, each has capabilities and drawbacks. While MACE has an intuitive interface and reliable performance, the research team was also using the MACE operator to control the artillery FDC and any enemy actions. Utilizing VBS4 would allow an operator to assume control of any position in the maneuver element, which could provide value for a gunner to work through shifting and ceasing fires in support of maneuver. To explore the problem space, the research team attempted to incorporate maneuver forces in both simulation systems.

In MACE, creating the maneuver forces and providing a series of waypoints for a movement track is very intuitive. The research team was able to easily create routes for the support by fire element and assault elements to follow. When the scenario reaches the point where the maneuver force should move from their starting position, simply changing the "route speed" property from 0 to 25km for the assault element and support by fire element causes them to move along their routes. With both elements in a "weapons free" status, the constructive entity behavior is to engage enemies identified within weapons range, which supports the scenario requirements.

In VBS4, there are two general options for planning and implementing constructive forces. The first method, using VBS4's Plan mode, allows a user to select from common pre-built units and quickly assign tactical tasks such as move, defend, assault, or suppress. Building a scheme of maneuver with these tasks is intuitive, and automatically calculates movement times to build an adjustable timeline for the scheme of maneuver. Selecting "build mission" in plan mode converts the general plan into detailed step-by-step instructions for VBS4's Control Artificial Intelligence (AI). This results in AI behavior that is realistic, responsive, and maintains an adequate level of presence for a training audience. The downside of this approach is that the AI is strictly tied to the linear timeline generated in Plan mode, preventing conditions-based or on-call changes or triggers. Although the Control AI does support customization and scripting, due to the timeline and scope of this program, the research team did not explore the possibilities of implementing more flexible tactical control into VBS4. Figure 28 displays the DIS integration of MACE and VBS4 entities using this method for maneuver forces.

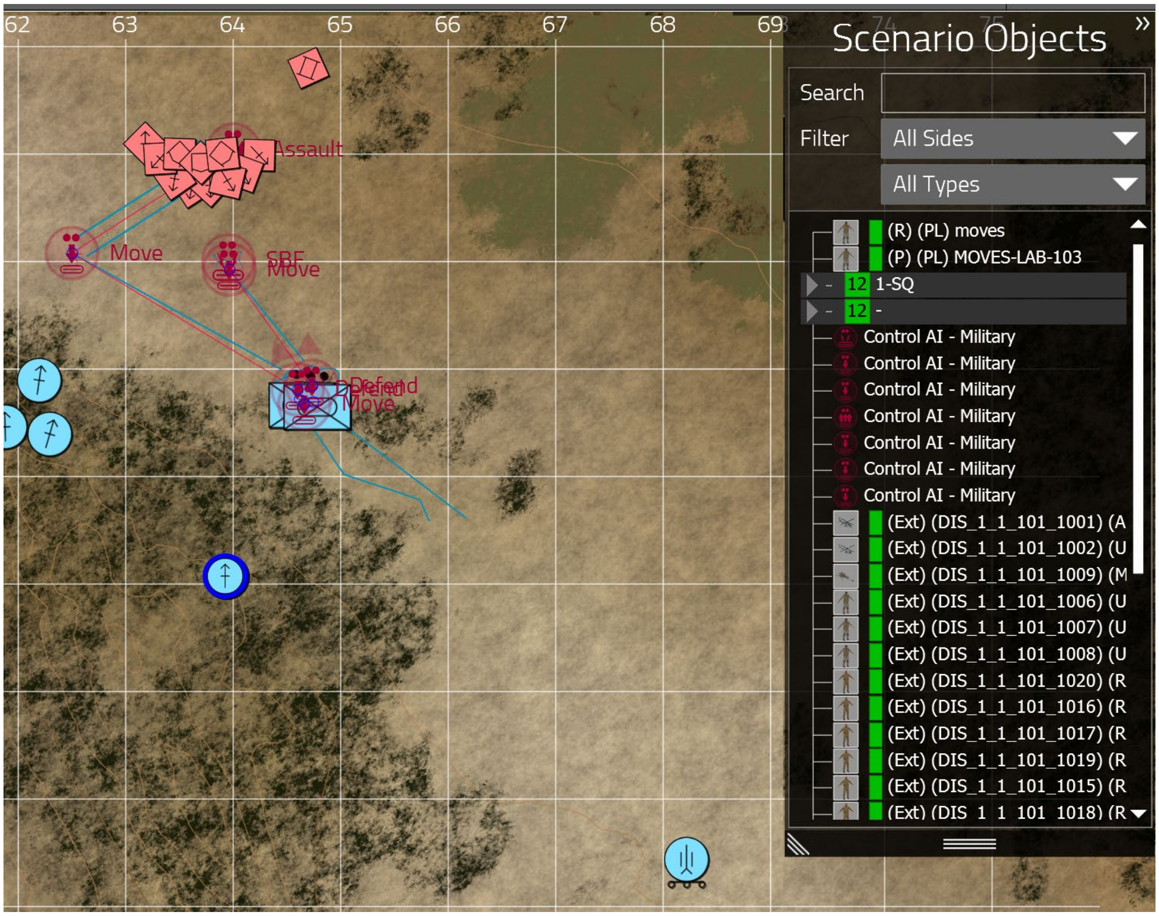


Figure 28. VBS4 View of Thread Testing with MACE DIS Integration

The second method for creating and controlling AI entities is more traditional for users of constructive simulations. Each entity or unit can be instantiated individually, and configured with parameters including orientation, speed, weapons status, formation, and more. Then a series of waypoints or objectives can be created for a movement path, with status changes at each waypoint. By creating a “hold” waypoint at the start of a path of “move” waypoints, the maneuver forces will hold their initial position when the scenario starts. When the scenario calls for maneuver forces to begin advancing, an administrator can simply delete the hold waypoint, and the AI elements will immediately begin moving to the next waypoint along the path. This method allows for mechanical movements and flexibility in timing, but it is harder to implement expected behaviors for suppression from the support by fire position.

The research team conducted thread testing with MACE and both VBS4 options controlling maneuver forces. The hardware requirements for VBS4 are very high, and even on an expensive high-end desktop purpose-built for graphical simulations (CPU, RAM, GPU), VBS4 performance was poor. In testing, when incoming artillery volleys began to land, VBS4 slowed or froze for 5–15 seconds. While frustrating, this might slow but does not prevent a Forward Observer from observing or communicating. However, when VBS4 controls maneuver forces, for the 5–15 seconds while frozen, the maneuver forces do not advance. This does cause two significant issues. First, the movement takes nearly twice as long, as these freezes occur every 30 seconds when a volley detonates. Second, because when a DIS simulation stops receiving updates on an entity, it uses dead reckoning to display the entity as continuing on its last trajectory. In this case, VBS4 will move the maneuver forces forward, then freeze. Other simulations will see the vehicles moving forward, and when DIS PDUs stop arriving from VBS4, show the vehicles as continuing on the same path and speed. When VBS4 unfreezes, it begins sending PDUs again, with the vehicles proceeding from the frozen location. On the other simulations, this snaps the vehicles backwards along their path. In effect, the other simulations show the vehicles moving on a path, but every 30 seconds they jump backwards to where they were 5–15 seconds ago. This will frustrate a training audience and kill any sense of presence or tactical verisimilitude. As a result, the research team decided to control the maneuver forces within MACE and accept the limited ability for a maneuver force trainee to directly control the vehicles movement or fires. In this case, when VBS4 freezes, the external scenario continues to progress and as VBS4 unfreezes, the operator returns to an expected state in the continued scenario.

O'Connor and LeSeur also include testing data collection for each of these threads in this activity block (2020). Because this process is designed to be applicable to any simulation environment, data collection can mean very different things. In the test and evaluation domain, data collection is the ultimate goal of the simulation environment, and ensuring the simulation environment provides the appropriate data accurately and in the manner expected is critically important. In the training domain, the simulation environment is intended to provide experience, but there are data outputs that can add value to the experience. After-action reviews (AAR) are expected after every training evolution, and simulation environments that can record pre-defined metrics or provide replay or summaries of friction points or key activities can add tremendous value. For this research project, our goal is a proof-of-concept demonstration, and capturing video of simulation screens, audio of radio traffic, and video or photographs of operator execution are desired. In this activity block, the research team ensured video capture worked for each simulation application, and that ASTi Simscribe captured radio traffic for playback.

With local integration and testing goals met for the key threads, the research team progressed to remote testing. Utilizing Windows Remote Desktop Connection (RDC) application, operators were able to connect to the ARGON VPN from home and control the computers in the MOVES Lab. Repeating the key threads remotely provided a few insights. For the ASTi Voisus software, the user must enable audio recording in RDC options, local resources, audio settings before connecting, as shown in Figure 29.

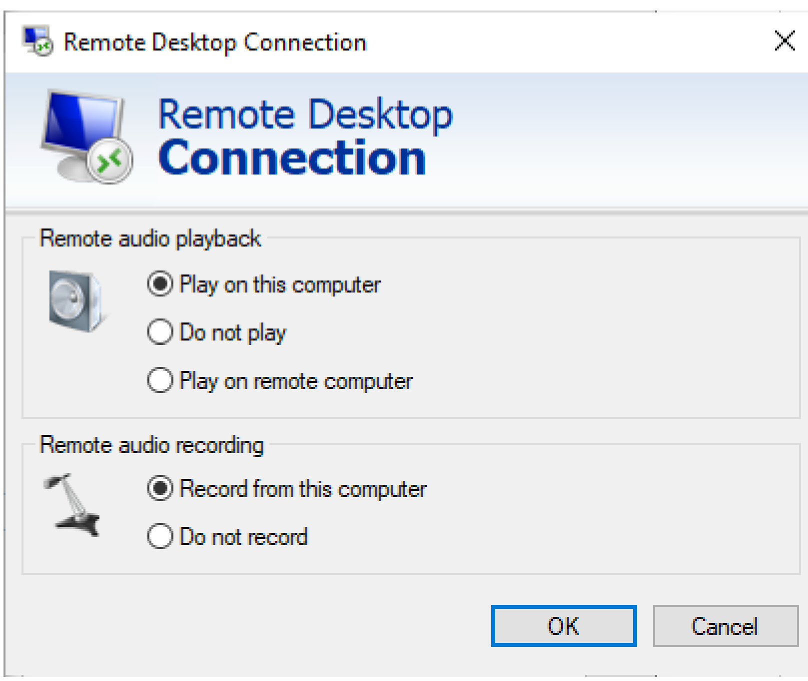


Figure 29. Remote Desktop Connection Audio Settings

Once connected, the research team took some time to configure audio settings before the VOIP audio was loud and clear. This included audio settings on the remote computers, MOVES Lab computers, and in the ASTi Voisus client. Utilizing the windows microphone audio settings to reduce input levels to 95 percent helped prevent audio input from garbling.

Remote Desktop Connection allows for control of a distant computer; but, depending on the connection and hardware users, should expect some choppiness and latency in both the visual display and the control inputs such as mouse and keyboard. The map and menu interface of MACE performed reasonably well from a remote operator. VBS4's choppy performance was not helped by latency between inputs and graphics across the RDC, but for a static observer oriented on a static objective and concerned with communication and coordination the performance was adequate. For PREPAR3D, the need for accurate, timely, and synchronized graphics and control to provide a reasonable flight simulation made RDC performance unsuitable. Additionally, translating the control inputs from the PUMA flight controller at the remote location to the simulation on the local computer in the MOVES Lab proved challenging. For these reasons, the research team decided to operate PREPAR3D locally inside the MOVES Lab for full scenario testing and execution. With the key threads integrated and tested locally and remotely, the research team was ready to move on to testing the complete scenario from start to finish.

d. Full Scenario Testing, Operator and Role Player Training, and Dry Runs

Because the operators and role players are the same in this scenario, and this simulation environment does not need additional support for execution, these four activity blocks were conducted concurrently. If this were a large-scale training exercise with separate system operators, role players such as a response cell, and a training audience such as a headquarters staff, these activity blocks would have to be progressively executed as resources were incrementally available to ensure the full scenario was ready for successful execution.

To ensure realism of scenario execution and credibility of subjective assessments, the simulation roles were assigned to volunteers with appropriate backgrounds. The

Artillery FDC was played by an 0802 Artillery Officer. The Fire Support Coordinator and Air Officer Roles within the FSCC were filled by a former C-130 Pilot with a Direct Air Support Center background, experienced in the routing and deconfliction of aircraft and fires. The rotary wing close air support pilot was a 7563 UH-1Y pilot with experience instructing other pilots during live-fire and simulated close air support training. The FiST Leader and Forward Observer roles were played by an 0302 Infantry Officer with FiST and FSCC experience. A second weapons and tactics officer qualified 7563 UH-1Y pilot—with close air support experience in combat, and instructional experience in live-fire and simulated close air support training—performed the role of the JTAC. The maneuver forces were portrayed by a second 0302 Infantry Officer. The experience of these Officers in operations and training, to include simulation training, provide a level of expertise sufficient for this research to establish face validity for this simulation environment.

Starting locally to reduce friction and increase cohesion, the research team familiarized each operator with their roles, system controls including ASTi Voisus and pertinent simulations, and the scenario MSEL. Then, the research team executed the complete scenario from start to finish, paying attention to friction points, timing and sequencing, and complete scenario integration. Friction points identified included some operators filling multiple simultaneous roles, timing and sequencing of fires, and synchronization of time. Minor adjustments and multiple repetitions conducted during dry runs allowed everyone to become comfortable, confident, and to communicate and execute effectively. With the entire scenario executed smoothly locally, remote testing was a relatively simple transition. Configuring audio settings to ensure clear communication via ASTi Voisus from remote connections proved to be the biggest friction point, but once complete, remote scenario execution went relatively smoothly after local rehearsals. Figure 30 displays the local testing configuration within the MOVES Lab.

From left to right: Artillery FDC, the FSCC, the CAS pilot, and an unmanned FiST station.

Figure 30. MOVES Lab Local Testing

With local and remote scenario testing and rehearsals complete, the research team demonstrated the capability of the simulation environment. For purposes of this project, the desire to record execution required only scheduling and coordination with the NPS media team for formally accomplishing DSEEP Step 6, Execute Simulation.

THIS PAGE INTENTIONALLY LEFT BLANK

V. EXECUTION AND ANALYSIS

A. DSEEP STEP 6. EXECUTE SIMULATION

"The purpose of this step of the DSEEP is to execute the integrated set of member applications (i.e., the 'simulation') and to preprocess the resulting output data" (IEEE Computer Society 2010a, 33). Depending upon the purpose and nature of the simulation environment, this step can be the culmination of the process. For a training simulation, this is where the simulation environment provides the training audience with the desired experience. For a test or analysis simulation, this step generates the desired data that can then be analyzed and evaluated. The two activities in this step are the actual execution of the simulation and preparing simulation environment outputs (IEEE Computer Society 2010a). For this project, the execution of the simulation was simply a scheduled repetition of the dry runs conducted in step 5. The preparation of simulation environment outputs involved a brief after action review as well as photo, video, audio, and screen recording of the scenario to develop a short demonstration video. To facilitate the media collection, this execution was conducted locally at the MOVES Lab, with operators physically separated to require radio communication for effective execution. Figures 31–33 display scenario execution.

The Artillery FDC (left) operates MACE to provide responsive and effective artillery fires for the simulation. The FSCC (right) provides routing and approval as the Air Officer and Fire Support Coordinator.

Figure 31. MACE Operator Station

The close air support helicopter is seen advancing from left to right toward the BMP targets, along with artillery detonations.

Figure 32. VBS4 FiST Perspective

Figure 33. PREPAR3D Operator Station

B. DSEEP STEP 7. ANALYZE DATA AND EVALUATE RESULTS

> The purpose of this step of the DSEEP is to analyze and evaluate the data acquired during the execution of the simulation environment (Step 6), and to report the results back to the user/sponsor. This evaluation is necessary to confirm that the simulation environment fully satisfies the requirements of the user/sponsor. The results are fed back to the user/sponsor so that they can decide if the original objectives have been met or if further work is required. In the latter case, it will be necessary to repeat some of the DSEEP steps again with modifications to the appropriate products. (IEEE Computer Society 2010a, 36)

Based upon the purpose, details, and desired outputs of the simulation environment, this step can vary greatly. DSEEP provides two activities; analyze data and evaluate and feedback results (IEEE Computer Society 2010a). This thesis is the development of a simulation environment for an intended purpose, not a quantified study of training effectiveness. The analysis of data involved a subjective assessment by the research team to determine if simulation environment objectives were met. The results are organized into suitability of each member application, simulation environment interoperability, simulation environment objectives, and recommendations for repeating or expanding this work.

1. Member Application Suitability

Each of the member applications was subjectively evaluated by the research team according to its effectiveness as a part-task trainer, interoperability with the simulation environment, and general ease of use or accessibility for a training audience.

a. ASTi Voisus

Central to the simulation environment, ASTi Voisus was the only application utilized by all operators and the only simulation interface for the FSC. It provided effective communications between operators, critical for the coordination of a fire support coordination exercise. Although setting the server up required some technical expertise, the ASTi support team was available, responsive, and very helpful in assisting and guiding the research team. Once the server was established, creating the comm plan and roles required for the scenario was intuitive, quick, and easy to adjust.

Operating the client software proved similarly intuitive, with minimalist but accessible user interfaces to keep training audiences focused on their tasks. With Windows Remote Desktop Connection properly configured, ASTi Voisus worked just as well from a remote VPN connection as locally in the MOVES Lab. The biggest challenges the research team encountered with ASTi Voisus were ensuring the headsets available worked properly and configuring audio and microphone settings within Windows as well as the ASTi client. Recording radio traffic for after-action reviews is available utilizing ASTI's Simscribe program, which was not licensed for this research project. In general, ASTi Voisus provided far more capability in radio configuration and modeling than was necessary for this simulation environment.

b. VBS4

Intended primarily as an image generator with access to optics, maps, and sensors to understand battlespace geometry and visualize friendly and enemy forces, fires, and effects, VBS4 proved generally effective. The library of available entities for inclusion into a scenario allows VBS4 to be used for a wide variety of tactical environments and training purposes. The optics and sensors, including far target location and laser designation, are particularly well suited for FiST training from a static observation post. The broad, general purpose capabilities of VBS4 exceeded the requirements of this research effort, but unfortunately were hampered by poor system performance. Even on a top-of-the-line computer purpose-built for graphical simulations—Intel Core i9-9900k CPU, Dual NVIDIA GeForce RTX 2080Ti GPUs, 32GB RAM—VBS4 regularly had issues with low framerate, sluggish responsiveness, and freezing when rendering detonations. Changing the graphics settings to medium or low did not improve performance noticeably. Due to time requirements, the research team was unable to resolve these performance issues or ascertain if the issues stem from VBS4's performance requirements, software configuration, or hardware issues that are not systemic or indicative of overall performance.

The VBS Gateway was powerful and intuitive in configuring, observing, and refining interoperability using DIS. It allowed for effective mapping between VBS entities and DIS enumerations and was the primary interface for the research team to ensure entities were implemented correctly in the simulation environment.

Without any VBS4 experience or expertise, the research team did face a significant learning curve in implementing and executing the scenario as intended. With a broad range of capabilities, the vast possibilities of entities and behaviors available require some expertise in efficiently implementing within a scenario. Once the scenario is established, training execution is relatively straightforward, particularly for current trainees who are familiar with digital environments and controls. With the minimal controls required for a FiST in a static observation post, very little operator training is required. Effective control of maneuver forces is more complex and would require additional operator training if desired. Ultimately, as long as the simulation environment has at least one experienced VBS4 administrator, training audiences do not require significant experience or training to execute this scenario.

c. PREPAR3D

A Lockheed Martin flight simulator, PREPAR3D's flight modeling was generally suitable for this simulation environment. Paired with Pro Flight Trainer's PUMA helicopter flight control system, the flight experience and aircraft control were judged to be of effective fidelity for this research effort. Unfortunately, for a close air support role in this simulation environment, PREPAR3D does not feature high fidelity sensors or weapons employment. Although there are opportunities to add a generic IR or night vision camera within PREPAR3D, they are very technical and complex to implement and do not provide the far target location and sensor capability regularly used for close air support. Similarly, although there are multiple weapons configurations available in PREPAR3D, the weapons control for precision guided munitions is heavily abstracted and does not add value for weapons employment training. Although PREPAR3D does not provide high-fidelity training for every part-task involved in close air support missions, it does provide a suitable opportunity for coordination with ground forces and JTACs in a common simulation environment. The ability of the JTAC to observe suppression and include calls for effective suppression in a SEAD mission was specifically noted as an important training opportunity that is absent in current simulation training.

Regarding interoperability, although PREPAR3D advertises DIS capabilities with a Professional Plus license, it is not correctly implemented in the current version, v5.1. Although this issue has been identified by PREPAR3D support to be fixed in the next version (Metel 2021), the research team had to purchase an additional license for version v4.5 to implement DIS within the time available. Once version v4.5 was installed and configured, enabling DIS was simple and quick, and fuzzy mapping ensured each external entity was displayed in some form. As a flight-focused simulation, not every entity, such as USMC AAVs, were modeled and implemented in PREPAR3D. It is possible to add models and DIS configurations within the simulation, but this required time and expertise beyond this research effort, so it was deemed suitable that friendly mechanized forces were distinguishable from enemy mechanized forces. Additionally, the adjudication of damage to ground vehicles was inconsistent. PREPAR3D seemed to provide options only for ground vehicles to be intact or destroyed and could not model incremental damage. This resulted in multiple occasions where the pilot was shown a destroyed target, while other simulations displayed the (damaged) targets as functional and dangerous.

As a serious flight simulator, PREPAR3D is developed for and marketed to pilots and student pilots and intended to recreate realism in the cockpit. This includes an assumption on the willingness of an operator to put time and effort into configuring scenarios and entities without an intuitive graphical user interface (GUI). Even for a professional helicopter pilot and flight simulator enthusiast, configuring PREPAR3D and implementing the scenario required a significant learning curve. System configuration and scenario implementation required some expertise beyond the flight and simulation experience of many pilots.

d. MACE

As a simulation providing constructive forces and fires across air and ground domains, and focused on the interaction across these domains, MACE proved very effective and powerful. The wide range of assets, capabilities, and detailed properties and controls are well balanced by intuitive operator interfaces. Ground entities, including indirect fire and air defense assets are easy to select, emplace, group and direct, and perform their roles without requiring direct operator prompting. Additional detailed interfaces for indirect fires and close air support allow the operator to quickly and intuitively input an observer's call for fire or close air support 9-line into pre-formatted fields and provide accurate readbacks and radio calls in execution. This includes adjusting fires based on observer corrections or identifying final attack headings based upon laser direction for precision guided munitions. As such, it does not require a full fire direction center or close air support pilot to provide requested fires but does allow either to work through the communication and coordination involved.

Interoperability via DIS was simple to execute with MACE, and any adjustments to DIS enumerations required were made in the VBS Gateway. No DIS configuration or entity mapping was required in MACE, so its capability and effectiveness in this regard was not explored by the research team.

As previously discussed, MACE's interface balances detailed and powerful capabilities with intuitive operation. Although a trained and experienced MACE operator would be capable of developing and executing more complex and varied scenarios, particularly with simultaneous engagements, a new operator would have few problems in developing and executing simple scenarios and progressively adding additional assets and capabilities.

With each member application evaluated, it is important to assess the simulation environment as a whole and the interoperability between member applications.

2. Simulation Environment Interoperability

On the local network, DIS was very easy to integrate. Ensuring DIS enumerations were correctly interpreted and implemented between simulations was the most laborious task, but tools such as DIS Spy and VBS Gateway were extremely helpful. DIS is not an ideal solution for remote network applications or large-scale simulation environments.

NPS' ARGON VPN or other VPNs allow for remote connectivity on distant networks but does not support UDP broadcast or multicast required for DIS implementation. Remote Desktop Connection allowed for remote users to control locally networked simulations using the VPN, but it has a detrimental effect on performance and latency. For a stationary observer, the degradation may be acceptable, but for interactive requirements such as a flight simulator, synchronization of visual output and control inputs are more concerning. Significantly, issues passing device inputs from joysticks or specialized flight controls from remote clients prevent increased fidelity for flight or vehicle simulation.

Timing and synchronization provided a few challenges for this simulation environment but were not insurmountable. Because DIS runs real time, each application maintained its own scenario timing that did not translate to other applications. This required the research team to manually synchronize watches, which is not unrealistic, but prevented the use of synchronized GPS time which would be available from many tools represented in the simulation environment. Additionally, this required additional cognitive load for the MACE operator to manually calculate time of flight and fire missions precisely on time to support a requested time-on-target.

There were no major disconnects identified in entity positioning, fires, or detonations. The biggest issue regarding entity interaction in the simulation environment was the varying interpretation of damage to vehicles. As previously mentioned, PREPAR3D interpreted ground vehicles as destroyed when the systems controlling them implemented them as damaged but still combat effective. This was particularly unexpected, as DIS demands every simulation defers to the owning simulation for an entity's status or effects of an interaction.

As described and illustrated in Chapter IV, terrain differences in the simulation environment provided more than cosmetic issues, as each member application implemented its own terrain. Big picture elevation data was largely consistent between simulations, which allowed for a common understanding of the macro-environment with recognizable landmarks that matched the real terrain, such as Quack Pass and the ridgeline with OP Creole. Detailed visualization of vegetation, texture, and micro-terrain varied between simulations. This proved problematic for target correlation between the JTAC and the pilot, as detailed visual references in VBS4 were not aligned with those in PREPAR3D. Major terrain features and marks such as smoke allowed for scenario execution, but training value for target correlation would be greatly increased by consistent visualization of terrain between applications.

3. Simulation Environment Goals

Having evaluated each member application, as well as the systemic interactions between them, it is important to assess the simulation environment as a whole against the stated goals. Table 7 captures the research team's evaluation of the simulation environment's ability to meet the training objectives defined in DSEEP Step 1.

Table 7. Evaluation of Simulation Environment and Training Objectives

Role	Training Objective	Component Requirements	Technical Requirements	Member Application	Evaluation
JTAC	Coordinate Close Air Support	Observe Targets, Friendly Forces, Aircraft Observe Fires and Effects Understand / Sense Battlespace Geometry and Timing Request Close Air Support Visual Target Correlation with Pilot Integration of CAS with Ground Fires and Maneuver	Image Generation GPS Far Target Location Radio with CAS Aircraft Coordination with FiST Virtual	VBS4 ASTi Voisus	Effective Image Generation Poor Performance During Fires Map, GPS, Compass, Optics Suitable Radio Communication Effective Terrain Differences Recommend Mark Generally Effective
FO	Coordinate Ground Fires	Observe Targets, Friendly Forces Observe Fires and Effects Understand / Sense Battlespace Geometry and Timing Call for Fire Adjust Ground Fires onto Target Integration of Ground Indirect Fires with CAS and Maneuver	Image Generation GPS Far Target Location Radio with FDC Coordination with FiST Virtual	VBS4 ASTi Voisus	Effective Image Generation Poor Performance During Fires Map, GPS, Compass, Optics Suitable Radio Communication Effective Suboptimal Performance, But Effective Generally Effective
FiST Leader	Integrate Fires in Support of Maneuver	Observe Targets, Friendly Forces, Aircraft Observe Fires and Effects Understand / Sense Battlespace Geometry and Timing Coordination between Maneuver Force and Firing Agencies Integration of CAS and Ground Fires with Maneuver	Image Generation Radio with Maneuver Radio with FSCC Coordination with JTAC, FO Virtual	VBS4 ASTi Voisus	Effective Image Generation Poor Performance During Fires Map, GPS, Compass, Optics Suitable Radio Communication Effective Generally Effective
CAS Pilot	Provide Close Air Support	Part-Task Training Fidelity Flight Controls Part-Task Training Fidelity Visual Perspective Observe Targets, Friendly Forces Observe Fires and Effects Understand / Sense Battlespace Geometry and Timing Coordinate Close Air Support Control Single Aircraft in Simulation Environment Employ Weapons Systems against Targets Visual Target Correlation with JTAC Conduct Close Air Support	Image Generation Flight Controls Flight Control Weapons Control Coordination with JTAC Virtual	PREPAR3D PFT PUMA ASTi Voisus	Very Effective Effective Image Generation In Cockpit Effective Image Generation Effective Image Generation Limited GPS, Map Understanding Very Effective Very Effective Extremely Low Fidelity Terrain Differences Require Mark Generally Effective
FDC	Coordinate Indirect Fire Support	Receive and Process Calls for Fire Provide Forward Observers with Firing Data Provide Ground Indirect Fires Provide Suppression or Mark for SEAD Missions Adjust Fires per FO Requests	Coordination with FO Fire Mission Control Message To Observer / Firing Data Constructive	MACE ASTi Voisus	Very Effective Very Effective Very Effective Very Effective Very Effective
FSCC	Process and Approve Fires	Monitor Communications from FiST, CAS, FDC, Maneuver	Coordination with FO, FDC, FiST, JTAC, AirO, Maneuver	ASTi Voisus	Very Effective
Maneuver	Execute Ground Maneuver	Observe Targets, Friendly Forces, Aircraft Observe Fires and Effects Understand / Sense Battlespace Geometry and Timing Maneuver in Simulation Environment Engage Targets with Direct Fire Weapons Coordinate CAS and Ground Indirect Fires through FiST	Coordination with FiST Vehicle Control Weapons Control Virtual or Constructive	MACE VBS4 ASTi Voisus	VBS4 Effective VBS4 Performs Poorly VBS4 And MACE Effective MACE Effective, VBS4 Disjointed Constructive, Not Virtual Generally Effective
Enemy	Ground Defense Air Defense	Identify and Destroy Friendly Maneuver (if unsupressed) Identify and Destroy Friendly Aircraft (if unsupressed)	Constructive	MACE	Very Effective Very Effective

Although there are many opportunities identified for improvement, the simulation environment is largely successful in meeting the defined objectives and can be repeated as necessary or refined as necessary to implement those improvements.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSIONS

A. SUMMARY

In general, the simulation environment was successful in providing a common scenario environment for fire support coordination along with some level of fidelity to support part-task training. Utilizing available software on unclassified hardware and networks, the research team was able to create a face-valid training environment. Although greater fidelity for close air support sensor and weapons employment—along with smoother performance for ground observers and maneuver forces—would have improved the task-training value, the simulation environment met the stated objectives. Although the *remote* training capability fell short of the initial target, with remote desktop capability and a *local* flight simulator, distributed training via a VPN is feasible. Based upon the experience gained during this project, the research team developed recommendations for repeating this simulation environment for training, as well as recommendations for future research projects. For any training use or future development in this domain, the research team highly recommends contacting the I MEF LVC-TE Team and starting with the baseline progress established with OASIS. They may be able to provide solutions and details regarding technical interoperability, training employment, and authority to employ simulation systems and products on Marine Corps hardware and networks.

B. RECOMMENDATIONS FOR TRAINING USE

If the research team were consulted in developing a similar simulation environment to support a specific training audience or use case, lessons learned from this thesis would drive recommendations for simulation system selection, simulation environment development, and network design.

For member application selection, the use of Program of Record simulation systems would be highly encouraged. In an academic research setting with a short time window, the simulation systems used were based upon availability, accessibility, and suitability. When a training audience has access to a validated simulation system that is authorized for training, that should be the default choice unless there are specific requirements that are not supported. OASIS is an exemplar of connecting existing and available hardware and software to create a simulation environment tailored to maximize the utility of existing simulation systems and integrate their capabilities.

Specifically, based upon the applications used in this project, the research team would seek to employ ASTi Voisus and MACE. VBS4 has a broad range of capabilities that were not implementable in this project due to performance issues, poor framerate, and program freezing. If VBS4 cannot perform adequately, VBS3 would be recommended instead, as it has been a program of record for many years, has expanded functionality including VBS Gateway, fires and strike programs, and is deployable on ubiquitous less expensive hardware systems. PREPAR3D provides good simulation for flight control and aircraft performance but is not primarily intended for realistic close air support training. If integration with program of record flight simulators for close air support or intelligence, surveillance, and reconnaissance platforms is an option, it is highly recommended.

Regardless of the simulation systems selected, a deliberate approach to terrain synchronization is recommended. The training value of target correlation between JTAC and pilot in a commonly represented simulation environment cannot be overstated. The research team did not adjust terrain in any simulation system for this project but would recommend a future environment develop commonly displayed terrain (elevation, vegetation, texture, roads, micro-terrain) for common training environments such as the Quackenbush corridor or Delta-T at 29 Palms. The effort in synchronizing terrain for commonly utilized training areas will generate reusable products that can be applied multiple times for different training events.

Finally, the network architecture is important to consider. If possible, it is recommended for both technical and training value to utilize a single training site with all systems on a local network. If this is not feasible, Remote Desktop Connection must either be configured to allow USB input to pass through for flight controls or other hardware, or those systems requiring direct inputs should be located on the local network. If interoperability between truly remote systems is required, the simulation team should understand the limitations of DIS, and seek to establish an HLA federation or other multi-architecture environment. The complexity, expertise, and effort required to successfully implement an HLA or multi-architecture federation across remote networks make it a valuable target for future research efforts.

C. RECOMMENDATIONS FOR FUTURE WORK

To expand and improve upon the capability developed with this simulation environment, a truly remote implementation is an ideal target. As discussed, this would likely require a dedicated VPN such as NPS' ARGON VPN, and an alternative to UDP broadcast communications. HLA is the obvious choice for a centrally managed simulation environment with a publish and subscribe methodology not requiring UDP broadcast or multicast capability. The challenge with HLA is to establish the federation and configure the runtime infrastructure (RTI) and member applications, which requires much more detailed configuration and verification than implementing only DIS. A number of tools, such as the Joint Staff J7 Exercise Division's Joint-LVC Simulation Protocol Analyzer (JSPA), have been recommended to identify simulation environment requirements and to interpret and translate both HLA and DIS traffic for a multi-architecture environment. A local DIS environment such as the one created for this project could potentially be augmented by a DIS bridge to an HLA Federation, allowing DIS-only applications to run locally, while HLA compliant applications could be executed remotely via VPN. This is broadly the architecture used by OASIS, which allows for a local enclave of VBS3 ground simulations stations and ASTi Voisus radio clients to interact with local and distant flight simulation systems on the ADVTE network. Replicating this architecture in a research environment utilizing available systems and unclassified networks could provide valuable information for future training systems such as the Marine Corps' LVC-TE.

Beyond improving remote access to the simulation environment, increasing interoperability with additional simulation systems would prove valuable. In an era where increasing Naval integration is prioritized, one desirable improvement would be to augment the simulation environment with simulated naval surface fires. This would allow U.S. Navy Shore Fire Control Parties (SFCP) to train integration with Marine Corps fires agencies, particularly the Air Naval Gunfire Liaison Companies (ANGLICO), but also including more ubiquitous FiSTs, FSCCs, close air support platforms, and ground indirect fires. A Naval Research Program Deckplate Innovation Study Challenge proposal was submitted in March 2021 to prompt research efforts into including naval surface fires into joint and Marine Corps fire support coordination simulation systems.

Another ideal target for expanding interoperability is into command and control systems. Although command and control channels at lower levels are historically restricted to voice radio, higher echelons currently employ computerized systems. Accessibility of digital systems, as well as speed, range, and bandwidth of data transmission are only expected to increase in coming years, suggesting newer command and control systems will provide value at lower levels than ever before. Integrating command and control systems into fire support coordination simulations is a valuable effort and will help current and future forces utilize simulation training while "training as they fight" with command and control architecture. Three current systems are attractive candidates for integration into the simulation environment: C2PC, AFATDS, and KILSWITCH.

C2PC, the official abbreviation for Command and Control Personal Computer, is the current Marine Corps program for command and control, and is utilized across the Marine Corps at the battalion or squadron level and above (Thome 2002). Integrating C2PC into the simulation environment would enable realistic training for battle staffs utilizing the same operational equipment they deploy with.

"The Advanced Field Artillery Tactical Data System (AFATDS) is the Fire Support Command and Control (C2) system employed by the U.S. Army and U.S. Marine Corps units to provide automated support for planning, coordinating, controlling and executing fires and effects" (Raytheon n.d.). AFATDS is currently used to develop and coordinate targeting and firing data solutions for Marine Corps artillery units during combat operations, live-fire training, non-fire rehearsals, and simulated training. AFATDS integration into the synthetic training environment could improve FDC training fidelity, but also integrate live artillery fire training with simulated maneuver and close air support. This would add particular value for units with limited live-fire training opportunities, such as those lacking suitable ranges, or those who are forward deployed and preparing for potential operations.

The Kinetic Integrated Lightweight SoftWare Individual Tactical Combat Handheld (KILSWITCH) application was developed by Naval Air Warfare Center Weapons Division, and allows pilots and ground controllers to use a handheld device to view objective area imagery with overlays for friendly and enemy positions, weapons delivery zones, and other tactical control measures (Rettedal 2016). JTACs using KILSWITCH linked to tactical radios can utilize the application to quickly develop close air support 9 lines, transmit them digitally to the aircraft, and allow the pilot to display graphical and textual information with position overlays (Rettedal 2016). Integrating this real-time command and control interface into the simulation environment will allow JTACs and close air support pilots the ability to gain situation awareness and utilize operational tools to coordinate within the simulation environment. Translating DIS position data for the CAS platform and JTAC into real-time locations in KILSWITCH would replicate the real-world GPS location services, and immediately expand training value.

Future system integration should consider the recently approved international Standard for Command and Control Systems—Simulation Systems Interoperation (C2SIM) to provide a common approach for exchange of orders and reports across the C2 systems and simulation systems (Simulation Interoperability Standards Organization 2020b). C2SIM also creates opportunities to introduce simulated or actual robotics and autonomous systems into the mix, moving toward training environments supporting future multi-domain or all-domain operations.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX. MASTER SCENARIO EVENT LIST AND SCRIPT

Line	Event	Actor	Trigger	Details	Call Net	To	Info
1	Comm Checks	All	Start	Comm			per RIO list
2	FiST, MNVR POSREP	FiST	Comms Checks Complete	Comm	Fires	Fires	posrep: GRID
3	RW Check In AirO	RW	Posreps Complete	Comm	TAD	AirO	Aircraft Number, Type, Position, Altitude, Ordnance, Time on Station, Remarks.
4	RW Push to FiST	AirO	RW Check In Complete	Comm	TAD	RW	
5	Arty FIRECAP	Arty	RW Pushed to FiST	Comm	Arty COF	Fires	Location, Altitude, Azimuth of Fire, Round counts (HE / WP / Illum) Wpns up, Firecap time
6	RW Check In FiST	RW	RW Pushed to FiST	Comm	TAD	JTAC	Aircraft Number, Type, Position, Altitude, Ordnance, Time on Station, Remarks.
7	FiST Acquire Targets	FiST	Posreps Complete	Target Acquisition	Fires	Fires	Target Description and Locations. Will advise with gameplan.
8	RW Situation Update	JTAC	RW Check In Complete	Comm	TAD	RW	Air threat, Ground threat, friendly Pos, Gun Target Line, proceed to HA. Advise when ready for gameplan 9 line.
PHASE2							
9	FiST Gameplan	Fist	Phase 1 Complete	Comm	Fires	Fires	SEAD - Arty Suppress ZSU. RW Kill BMPs from BP, Stay West.
10	RW Push to HA	RW	Phase 1 Complete	RW push to HA	TAD	JTAC	RW Pushing HA, ready for gameplan 9 line
11	CAS gameplan + 9 Line	JTAC	Fist Gameplan Complete	Comm	TAD	RW	Gameplan and 9 line per Air SMEs. With Readbacks. STAY WEST. Sync TOT / TTT.
12	SEAD CFF	FO	Fist Gameplan Complete	Comm CFF	Arty COF	Arty	SEAD. Grid to Suppress ZSU. Grid to Mark BMP. Target to Suppress ZSU 23-4. Continuous (or nonstandard - provide timing) CAS TOT / TTT.
13	SEAD MTO	Arty	CFF Complete + 30	Comm MTO	Arty COF	FO	MTO - units to suppress, units to mark. 1 round in effect. **Target Number.** Time of Flight. Max Ord. GTL. Timing. TOT.
14	SEAD Approval	Fires	MTO Complete	Comm Approval	Arty COF	Arty	Target Number approved.
15	Arty Shot Suppression	Arty	TOT - (TOF + Suppression lead)	Arty Suppression	Arty COF	FO	Target Number Shot Suppression. Splash.
16	Arty Shot Mark	Arty	TOT - (TOF + 45)	Arty Mark	Arty COF	FO	Target Number Shot Mark. Splash Mark.
17	GTL Hot	JTAC	Ln15	Comm	TAD	RW	Rounds in the Air
18	RW attack sequence	RW	TOT - required time	RW attack sequence	TAD	JTAC	All calls between JTAC/RW. Pushing-continue, visual suppression and mark? Correction from mark. Inbound - cleared hot. Egress. BDA.
19	Arty Rounds Complete	Arty	TOT + Suppression tail	Comm	Arty COF	FO	Target Number Rounds Complete
20	BDA	FiST	SEAD Complete	Comm	Fires	Fires	BDA - effects. BMPs destroyed. Reattack if required?
21	Reattack (IF REQD)	FO/JTAC	Reattack, TTT 5	Comm	COF/TAD	RW/Arty	Coordinate reattack with TTT 5 minutes.
22	RW Egress	RW	Good BDA	RWs Egress to HA	TAD	JTAC	RW pushing HA, TOS remaining.
23	End of Mission	FO	Good BDA	Comm	Arty COF	Arty	Target Number - End of Mission. Echoed by Arty.
Phase3							
23	Mnvr Ready	Mnvr	SEAD Complete	Comm	TAC	Fires/FiST	Maneuver ready for closure when suppression effective.
24	CFF Adjust	FO	Mnvr Ready	Comm	Arty COF	FO	Adjust Fire to Impact defensive position. Will follow with duration suppression.
	Adjust MTO	Arty	CFF Complete + 30	Comm	Arty COF	FO	MTO. 1 round in effect. Target Number. Time of Flight. Max Ord. GTL.
	Adjust Approval	Fires	MTO Complete	Comm	Arty COF	Arty	Target Number Approved.
25	Arty Adjust	Arty	CFF Complete	Arty Adjust fire onto target	Arty COF	Arty	Target Number Shot Over. Splash Over. - One round 300m off target. FO corrections one round at a time to get closer to target.
	Observer Correction	FO	Impact	Corrections for Adjust Fire	Arty COF	FO	Target Number Add / Drop, Left / Right in meters. - repeat until on target, then call duration suppression X mikes.
26	Duration Suppression	Arty	Duration Requested	1 round x 6 tubes every 30s	Arty COF	Arty	Shot over. Splash over.
27	Suppression Effective	FiST	Suppression Impacts	Comm	TAC	Mnvr	Suppression Effective. PRO-WORD.
28	Mnvr OM	Mnvr	Suppression Effective	Start Moving to SBF and AP	TAC	Fires	
29	SBF Occupied	SBF	SBF Occupied	Comm	TAC	FiST	SBF Occupied at GRID.
30	Aslt past AP	Aslt	Turn at AP	Comm	TAC	FiST	Aslt at Aslt Pos, pushing to objective. POSREP.
31	Aslt PL Red	Aslt	Phase Line Red	Comm	TAC	FiST/SBF	Aslt at PL Red
32	SBF Suppress	SBF	PL Red Called	SBF Suppress Obj	TAC	Aslt	SBF beginning suppression
33	Cease Load Call	FO	DF Suppression (PL Red)	Comm	Arty COF	Arty	Target Number Cease Load.
	Arty Cease Load	Arty	Cease Load Called	Arty Stop Suppression Mission.	Arty COF	FO	Target Number Cease Load.
	End of Mission	FO	Cease Load Confirmed	Comm	Arty COF	Arty	End Of Mission. Mech infantry platoon suppressed.
34	SBF Cease	SBF	Aslt w/in 15 degrees	SBF Cease Fire	TAC	Aslt	SBF Ceased
35	Clear Obj	Aslt	Approaching Obj	Shoot any enemies left	TAC	Aslt	Aslt on objective. - Once clear, PROWORD, objective secured.
36	Establish Defense	Aslt	Obj Cleared	Move to Defensive position	TAC	SBF	Establishing defense
	Consolidate	SBF	Obj Cleared	Move to Defensive position	TAC	Aslt	Hasty Defense established
Phase4							
37	Counterattack	Enemy		Enemy formation approach N-S			
38	CAtk spotted	Mnvr/Fist	CATK identified	Comm	TAC	Fires	Enemy Armor column approaching N-S. Lead trace approx.
	FiST Gameplan	FiST	CATK id'd	Comm	Fires	Fires	Type 3 CAS from BP.
	Gameplan 9 line	JTAC		Comm	TAD	RW	Gameplan and 9 line per Air SMEs. Type 3. Keep fires and effects N of .... Arty cold.
	RW CAS Type 3	RW	cleared type 3.	RW Kill enemy formation	TAD	JTAC	

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Department of Defense. 2011. "Modeling and Simulation (M&S) Glossary." https://www.acqnotes.com/Attachments/DOD%20M&S%20Glossary%201%20Oct%2011.pdf.

———. 2016. "TENA Architecture Reference Document." https://www.trmc.osd.mil/.

Donaldson, Michael. 2021. "Synthetic Training Integration & Management Branch Command Brief." April 15.

FlightSafety International. 2018. "The Naval Air Warfare Center Training Systems Division Selects FlightSafety International to Build New Flight Training Devices." 2018. https://news.flightsafety.com/pressrelease/naval-air-warfare-center-training-systems-division-selects-flightsafety-international-build-new-flight-training-devices/.

Getchell, Chris. 2020a. "OASIS Communications Diagram." I MEF LVC TE.

———. 2020b. "RE: Networked Comms (for MOVES Thesis)," September 9, 2020.

IEEE Computer Society. 2010a. "IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process (DSEEP)." Simulation Interoperability Standards Organization. IEEE Std 1730–2010.

———. 2010b. "IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) —Framework and Rules." IEEE Std 1516–2010.

———. 2012. "IEEE Standard for Distributed Interactive Simulation—Application Protocols." IEEE Std 1278.1-2012.

———. 2013. "IEEE Recommended Practice for Distributed Simulation Engineering and Execution Process Multi-Architecture Overlay (DMAO)." IEEE Std 1730.1-2013.

Knight, Steven. 1998. "A Comparison of Analysis in DIS and HLA." Naval Postgraduate School. https://calhoun.nps.edu/handle/10945/8115.

Metel, Chris. 2021. "DIS Connection." *Prepar3D Forums*. https://www.prepar3d.com/forum/viewtopic.php?f=6323&t=141469.

Pro Flight Trainer. n.d. "Puma RTF Packaging (Ready To Fly)." https://pro-flight-trainer-com.myshopify.com/collections/control-full-sets/products/copy-of-pro-flight-trainer-4th-generation-puma-pas-packaging-pre-assembled.

Program Manager Training Systems. 2016. "Supporting Arms Virtual Trainer." http://www.tjinc-eng.com/SAVT_TRIFOLD.pdf 2016.

Raytheon. n.d. “Advanced Field Artillery Tactical Data System (AFATDS).” Accessed May 12, 2021. https://www.raytheon.com/capabilities/products/afatds.

Rettedal, Daniel. 2016. “Digital Interoperability in the Objective Area.” *Marine Corps Gazette* 100 (4): 50–54.

Simulation Interoperability Standards Organization. 2020a. “Reference for Enumerations for Simulation Interoperability.” SISO-REF-010-2020. https://www.sisostds.org/DigitalLibrary.aspx?Command=Core_Download&EntryId=51787.

———. 2020b. “Standard for Command and Control Systems—Simulation Systems Interoperation.” SISO-STD-019-2020. https://www.sisostds.org/DesktopModules/Bring2mind/DMX/API/Entries/Download?Command=Core_Download&EntryId=51771&PortalId=0&TabId=105.

Steed, Anthony, and Manuel Oliveira. 2010. *Networked Graphics*. Morgan Kaufmann.

Thome, Geoffrey. 2002. “U.S. Marine Specific Software Interoperability Requirements of the AFATDS and IOS Software Suites.” Naval Postgraduate School. https://ntrl.ntis.gov/NTRL/dashboard/searchResults/titleDetail/ADA407166.xhtml.

Tolk, Andreas. 2018. “The Elusiveness of Simulation Interoperability—What Is Different from Other Interoperability Domains?” In *Proceedings of the 2018 Winter Simulation Conference*.

Trott, Kevin. 2003. “Command, Control, Communications, Intelligence, Surveillance, and Reconnaissance (C4ISR) Modeling and Simulation Using Joint Semi-Automated Forces (JSAF).” Final Technical Report AFRL-IF-RS-TR-2003-144. Air Force Research Laboratory. June 2003. https://apps.dtic.mil/dtic/tr/fulltext/u2/a417477.pdf.

U.S. Marine Corps. 2016. “Unit Training Management Guide.” Department of the Navy. MCTP 8-10A. https://www.marines.mil/News/Publications/MCPEL/Electronic-Library-Display/Article/1120726/mctp-8-10a/.

———. 2017. “2017 Marine Aviation Plan.” https://www.aviation.marines.mil/Portals/11/2017%20MARINE%20AVIATIOIN%20PLAN.pdf.

———. 2018a. “Fire Support Coordination in the Ground Combat Element.” Department of the Navy. MCTP 3-10F. https://www.marines.mil/News/Publications/MCPEL/Electronic-Library-Display/Article/899543/mctp-3-10f-formerly-mcwp-3-16/.

———. 2018b. “Warfighting.” Department of the Navy. MCDP 1. https://www.marines.mil/portals/1/Publications/MCDP%201%20Warfighting%20GN.pdf?ver=2019-01-31-110543-300.

Vaught, Bryan. 2016. “21st Century Digital Battlefields: MAGTF Simulation Effectiveness for Training Marines.” Marine Corps University.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California