NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

CLEARED HOT: A FORWARD AIR CONTROL (AIRBORNE) CONCEPTS TRAINER

by

Gregory King
Charles Lakey

September 2006

Thesis Advisor:	Rudolph Darken
Co-Advisor:	Joseph Sullivan

This thesis done in cooperation with the MOVES Institute.
Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

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

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	September 2006	Master's Thesis

4. TITLE AND SUBTITLE Cleared Hot: A Forward Air Control (Airborne) Concepts Trainer

5. FUNDING NUMBERS

6. AUTHOR(S) Gregory King and Charles Lakey

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)
Virtual Technologies and Environments (VIRTE)
Office of Naval Research
Science and Technology Division

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
With the aim of creating a skill trainer of conceptual knowledge, what is the development process for ensuring the correct set of objectives are determined, matched to appropriate technology, and implemented? Months and years prior to the first instance of trainer use, the initial steps of the developer determine the end product's success. Computer based trainers fielded for use by the military are rife with poorly matched tasks to technology, often the product of contracts that begin with a list of high-level objectives imitating a detailed requirements document. In those cases, software developers are forced to make best guesses about how to meet those objectives. Is there a better method? We embarked on a project to create a trainer for the military aviation mission of Forward Air Control (Airborne) using a development process that first identified critical tasks, then matched technology to facilitate training those tasks, and finally allowed expert evaluation of positive transfer. We do not assume that our methodology—which foregoes a comprehensive transfer study—is the preferred approach; rather, in cases where such a study is not feasible, we assert that a good development process, reinforced with subsequent expert evaluation, is a comparable alternative.

14. SUBJECT TERMS
Forward Air Control(Airborne), FAC(A), Training, Open Source, Close Air Support, Virtual Environment, Call For Fire

15. NUMBER OF PAGES
229

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	UL

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

CLEARED HOT:
A FORWARD AIR CONTROL (AIRBORNE) CONCEPTS TRAINER

Gregory W. King
Lieutenant Colonel, United States Marine Corps
B.A., Morehouse College, 1990

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

Charles B. Lakey
Major, United States Marine Corps
B.A., University of Oklahoma, 1993

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN MODELING, VIRTUAL ENVIRONMENTS, AND SIMULATION (MOVES)

from the

NAVAL POSTGRADUATE SCHOOL
September 2006

Authors:	Gregory W. King
	Charles B. Lakey
Approved by:	Rudolph P. Darken Thesis Advisor
	Joseph A. Sullivan Co-Advisor
	Rudolph P. Darken Chairman, MOVES Department
	Peter J. Denning Chairman, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

With the aim of creating a skill trainer of conceptual knowledge, what is the development process for ensuring the correct set of objectives are determined, matched to appropriate technology, and implemented? Months and years prior to the first instance of trainer use, the initial steps of the developer determine the end product's success. Computer based trainers fielded for use by the military are rife with poorly matched tasks to technology, often the product of contracts that begin with a list of high-level objectives imitating a detailed requirements document. In those cases, software developers are forced to make best guesses about how to meet those objectives. Is there a better method? We embarked on a project to create a trainer for the military aviation mission of Forward Air Control (Airborne) using a development process that first identified critical tasks, then matched technology to facilitate training those tasks, and finally allowed expert evaluation of positive transfer. We do not assume that our methodology—which foregoes a comprehensive transfer study—is the preferred approach; rather, in cases where such a study is not feasible, we assert that a good development process, reinforced with subsequent expert evaluation, is a comparable alternative.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

We would like to take this opportunity to thank many individuals who, through their encouragement, advice, support, and when necessary, prodding, have made the completion of this thesis possible. First and foremost is the team of software engineers and graphic artists led by Erik Johnson. Jon Grant, Chris Osborne, Brad Anderegg, William Clapham, Mike Dunhour, and for a short time, Matt Pritchard, by committing thousands of man-hours to the Cleared Hot project, enabled what began as a vision to bridge a training gap to become functioning software. To Commander Joseph Sullivan, who believed in us even when we lost faith, we owe a large debt. His open door and timely rudder steers kept the project moving in the right direction. Dr. Rudolph Darken kept us from becoming lost in the minutiae and pulled us back to see the big picture on many occasions.

There were, in addition, many fellow officers more versed in Forward Air Control and organizing fire missions than us, whom we consulted in order to ensure trainer fidelity. You are too numerous to mention; know that you have our gratitude, as *Cleared Hot* would not have the sparkle of authenticity without your assistance.

Finally, we would like to thank our wives, Sharon and Diane, for their patience and understanding during our time here at the Naval Postgraduate School. The completion of our Master's Degrees would have been impossible without your sacrifice and support.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. MISSION

Forward Air Control (Airborne), or FAC(A), is a mission qualification acquired by military pilots of specific aircraft types. Within the United States Marine Corps, this includes the UH-1N, AH-1W, and F/A-18D aircraft. Once qualified, the pilots conduct airborne duties that involve coordinating the fires of air, ground, and sea-based weapon systems. Defined by Marine Aviation Weapons and Tactics Squadron One (MAWTS-1), the FAC(A) is

> ...a specifically trained and qualified aviation officer who exercises control from the air of aircraft engaged in [Close Air Support (CAS)] of ground troops...FAC(A) tasks include detecting and destroying enemy targets, coordinating target marking, providing terminal attack control of CAS missions, conducting air reconnaissance, providing artillery and naval gunfire air spotting, providing radio relay for the [Tactical Air Control Party (TACP)] or [Joint Terminal Attack Controller (JTAC)], and passing [Battle Damage Assessment (BDA)].[1]

Success at the FAC(A) mission relies on understanding the current situation on the battlefield as well as what is likely to develop. It requires the ability to visualize the geometry of fire support coordination measures (FSCM) and the movement of units through a three-dimensional target area. Furthermore, a FAC(A) must be able to speak the precise language of controlling fires in order to convey his plan to others. Tantamount to a juggling act, the job requires doing all this according to a timeline that allows very few deviations. The clock is always ticking for supporting aircraft due to fuel limitations, and adversary units may move quickly out of preplanned target areas.

Directing the fires of a single support unit onto a target is not a difficult task in and of itself; if one has sufficient time to calculate heading, distance, and elevation between the two entities, it is a simple matter of reading a doctrinal format while filling in the blanks with the data particular to the situation.

The enemy does not field its assets without their own protection, however. Air defense weapons in the form of Surface-to-Air Missiles (SAM) and Anti-Aircraft Artillery (AAA) pose a threat that must be honored. A primary challenge of the FAC(A) mission lies in ensuring that active enemy air defenses are suppressed while friendly attacking aircraft do their job. This is the essence of creating an attack package; it entails building a sequence of fires originating from multiple assets such that the enemy cannot effectively react. Arranging for indirect fire to shell enemy air defense to prevent it from firing on inbound friendly aircraft requires coordination and detail; friendly Close Air Support (CAS) may be flying above or below the trajectories of the indirect rounds, may need to stay abeam of certain terrain features described over the radio, and will have a specific time window for their attacks. It is the responsibility of the FAC(A) to set up all the moving parts and ensure the players understand their roles. Fratricide is always a present risk. If any of the elements fail, especially when friendly ground troops are fighting in proximity to the targets, the potential for friendly casualties greatly increases.

B. MOTIVATION

Within the U.S. Marine Corps helicopter community, the training requirements for obtaining the FAC(A) qualification are prescribed by MCO P3500.48A, the Aviation Training and Readiness (T&R) Manual for AH-1 pilots, and its counterpart, MCO P3500.49A, the T&R for UH-1 pilots. They dictate that pilots under instruction shall complete academic training in the form of an Expeditionary Warfare Training Group (EWTG) developed FAC(A) syllabus, followed by four syllabus sorties.[2,3] The EWTG ground school is comprehensive; it is a one week course designed to provide the requisite knowledge to create and execute attack packages. Among the critical subjects taught during the course are *Controlling CAS as a FAC(A)*, *Fratricide*, *Targeting*, *Fire Support Planning and Scheduling*, and *Fire Support Integration*. In addition, the course instructors lead several practical application exercises involving Fire Support Integration, and students spend time on a Forward Observer Training Simulator practicing Call For Fire (CFF).[4]

Yet for some there exists a training gap. Occasionally owing to budget constraints, at other times due to the operational tempo of a working squadron, pilots preparing to enter the FAC(A) syllabus are not afforded the opportunity to attend the EWTG course. Nor is the school co-located with east- or west-coast Marine Corps squadrons; taking the course removes a pilot from flight duties for one week, interrupting mission currency windows. For one-third of the FAC(A)-capable squadrons, it also involves flying the pilots cross-country to attend the course. In these cases where the prescribed syllabus is either not available or not feasible, pilots are expected to research the applicable publications to acquire the basic knowledge needed to enter the FAC(A) syllabus. The respective airframe Tactical Manuals provide a wealth of knowledge to use in preparation, as do the doctrinal texts such as *Joint Tactics, Techniques, and Procedures for Close Air Support* and *Multi-Service Procedures for the Joint Application of Firepower*. The MAWTS handbook on FAC(A) is an especially valuable resource, consolidating all aspects of conducting the mission from planning to execution. As excellent as the written resources may be, however, they cannot provide an environment for experimentation. A percentage of pilots go from a self-study program directly into the cockpit for training mission execution.

The duration of each FAC(A) syllabus sortie averages two flight hours.[5,6] Armed with schoolhouse knowledge, and sometimes without it, presumably also with a generous portion of advice from senior squadron pilots, FAC(A) pilots-in-training find themselves at the center of a maelstrom of activity once they are airborne. CAS aircraft require attack plans sent according to a specific format, as do indirect fire agencies. The attacks may be preplanned, and should be so to the maximum extent possible, but they inevitably require tailoring once airborne. Target positions are not always known and unforeseen obstacles arise, ranging from radio transmission difficulties to equipment malfunctions and late-arriving aircraft. The FAC(A) must remain flexible; the seeming chaos is often a source of frustration and challenge for the uninitiated. Pressure to succeed in an uncertain situation is heightened by the presence of an instructor pilot evaluating the sortie.

Mistakes inevitably made during this time are costly in terms of fuel, ordnance, and man-hours for all involved parties. Fixed-wing assets are airborne at this time as well; typically two or more sections (increasing their own readiness in CAS) fly in support of the FAC(A) training sortie. Sequencing ground and air support, visualizing the trajectories and effects of their ordnance, and speaking the particular language of controlling fires is partly art, partly logic. The syllabus pilot who cannot grasp this interaction sacrifices valuable time while airborne. The “learning curve” is extremely steep during these sorties. Situational awareness is bought at a dear price; only following many such missions do pilots posses the experience to efficiently conduct Forward Air Control. Four sorties do not create a superior FAC(A), but four sorties is the allowance; flight hours are based on fuel budgets.

C. RESEARCH QUESTIONS

Is there room for improvement in the training of the FAC(A) mission? Who is qualified to answer this question? We took a cue from both our personal experiences as Marine Corps aviators and Forward Air Controllers as well as from interviews of senior flight instructors in Marine Corps AH-1W/UH-1N squadrons. Typical comments included that FAC(A) syllabus pilots were not able to fully absorb all that the instructors wanted to teach them while airborne. They spoke of opportunities arising to demonstrate advanced techniques for sequencing attacks, but the syllabus pilots were not yet familiar with conducting the basics and so struggled with these varsity level maneuvers. Confidence is bred from executing the same mission again and again, and given limited flight hours, that redundancy is missing.

Assuming that it is desirable to increase pilot understanding of the mission prior to beginning the syllabus flights, what are the possible venues for training? Wargaming in its many forms traditionally has been effective as a training tool[7]; it allows experimentation. Players execute a plan and see the reaction it draws from others. Sand table exercises work very well for this purpose, and facilities such as the Combined Arms Staff Trainer (CAST) in Twentynine Palms, California, take advantage of the rewards offered by pitting tacticians against one another.

The advance of technology has afforded us immersive worlds in which to practice our wargaming. Combat scenario simulations provide a safe and cost-effective means for honing tactics, and indeed, the Marine Corps has often received impetus from its most senior officers to find new ways to accomplish more training with less traditional resources. Lieutenant General W.L. Nyland, a recent Deputy Commandant for Aviation, devoted a chapter to the discussion of the impact of simulation on training in the 2002 Marine Aviation Campaign Plan, and an excerpt from that section illuminates his guidance:

> The focus of our simulator procurement is to challenge aircrews in a realistic flight environment with visual and mental stressors. The MCASMP (Marine Corps Simulator Master Plan) is designed to procure simulators that will provide a source of realistic flight experience to the already capable aviator. Although simulation is never a substitute for actual flight, it has immense value in the augmentation of aircrew training. These simulators will provide an unsurpassed enhancement to the valuable time an aviator spends in an aircraft. Our drive to increase simulator use reflects the worldwide trend in both defense and industry for this type of training. It also makes sense to save service life on our aircraft and decrease some of the risk inherent in certain operational and training flight scenarios.
>
> In certain instances, such as weapons proficiency and high threat scenarios, we will conduct realistic and economical training by utilizing a simulator rather than an aircraft. Simulator technology is rapidly improving, allowing aircrews and controllers to efficiently rehearse "real world" missions as a prelude to combat flight operations.
>
> Currently, many of our communities do not have adequate simulators either in quality or quantity....Commanders at all levels need to continue to emphasize simulator training as access to high fidelity simulation improves. We must embrace the valuable role that simulation will play in the future of Marine Aviation training.[8]

Clearly simulation is one means to bolster a FAC(A) syllabus pilot's preparation for the training flights. However, we were leery of using technology for technology's sake. It is often the case that, when a new capability appears, it is met immediately with plans to use it for training, when the training is fulfilled more cheaply and efficiently through other means. As an example, formation flights are often rehearsed by pilots walking around a parking lot, assuming the physical role of their aircraft. No computers are needed; the only virtual environment required is a spacious area. Practicing their spacing and coordinated turns with other members of the flight while on the ground—without the need to multitask other flight duties—results in a better understanding of formation dynamics.

In addition, it is neither feasible nor desirable to simulate completely the flights conducted in an aircraft. However, it would be beneficial to create a ground-based trainer to help solidify the understanding of how fire support platforms and their ordnance integrate in three dimensional space. Using such a trainer, a pilot would become familiar with the flow of a FAC(A) mission prior to being evaluated in the aircraft itself.

The problem of understanding ordnance trajectories and how they affect the numerous players on the battlefield has traditionally been one of the more difficult obstacles to overcome for FAC(A) syllabus pilots. Much time during an initial training flight is wasted due to the "newness" of having to visualize how key elements work together. If a trainer designed to explain and evaluate understanding of these concepts could be created, and if its fidelity proved sufficient, it is conceivable that it could be used to supplement the initial portion of the FAC(A) training syllabus. Acquiring a better understanding of the FAC(A) process prior to flight would increase a student's performance at the actual task. The question of skill improvement is not *if*, but *by how much and how effectively*.

Simulators provide a positive transfer of knowledge when they are designed and implemented with clear training goals in mind. In order to define these training goals, we need to examine the mission of FAC(A) to determine what are the skills we seek to teach. What type of knowledge do we hope to pass to the trainee – declarative, procedural, or conceptual? At what stage of proficiency will pilots enter the simulation training? Is there benefit in pilots training together with other representative players on the battlefield, training alone, or using a combination of both modes?

D. ROADMAP

It was our goal to find a conceptual knowledge training system for the mission of FAC(A), and failing that, to create one ourselves. We found it necessary to answer each of the previous questions and many more before we were certain simulation could provide the needed training. Our approach centered on analyzing the mission and singling out the critical tasks. We sought a training solution that would be of no cost to the military, was portable, and could train pilots individually. Furthermore, we required validation of the system, according to standards provided by pilots considered experts in the FAC(A) mission. This thesis is the documentation of that process.

We drew up a roadmap of the work to be done, beginning with analyzing the task of Forward Air Control. Chapter II describes our selection of a task analysis tool and identifies the critical tasks to be trained. From the beginning we knew the trainer would not attempt to simulate the mission in its entirety; partial task training would be key to keeping the trainer within manageable bounds.

Once the mission critical tasks were defined, we needed to look at existing systems. It was possible that one or several fulfilled our training requirements, although it was likely that none of them were being offered to the military at no cost. Chapter III presents a representative sample of trainers that focus on Forward Air Control, and illustrates where these systems meet or fail to meet our criteria.

Having found no systems that fulfilled all of our training requirements, it became our task to create one. Chapter IV is the story of development; it describes our process of moving from a list of critical tasks to fielding a prototype computer-based trainer. It explains our reasoning for choosing technology as a vehicle for the trainer, based upon our resources and ability to mitigate several negative factors associated with existing systems. We describe the team that worked on the project, the techniques we used, and certain design decisions made along the way.

Chapter V takes a more technical slant; it provides an overview of system implementation. In broad strokes, we discuss the architecture of the trainer. Working within an open-source game engine, agents within the trainer utilize a planning system to choose their actions. These “intelligent” units comprise the contingent of CAS aircraft which are directed in the formation of attack packages.

Validation of any trainer is critical; otherwise, there exists no proof that it has added any value to the training process. A responsibility lies with system developers to test their systems and publish results. Knowing this, it disappointed us greatly to bring our work on the project to closure with no such study of effectiveness. Two obstacles prevented current validation through an experiment. The first was that although *Cleared Hot* is functional software, it incorporates few of the debriefing tools from its original design, and furthermore lacks some features dealing with multiple types of indirect fire missions. The second obstacle was being denied access to the required participants. It is our hope that those who follow in *Cleared Hot* development will be afforded the chance to evaluate the trainer's effectiveness properly, and so Chapter VI outlines an experimental protocol.

Having failed to produce a system viable for testing with participants at the end of an 18-month development cycle, we did the next best thing: *Cleared Hot* was taken to the mission subject matter experts for evaluation. Chapter VII details our visit to MAWTS-1, their feedback on the trainer, and our conclusions on how it might be tempered to become a more effective training tool.

We've had to compress the number of features that made it into the final version of *Cleared Hot*. In most cases, functionality was restricted when a feature did not map onto a critical task. In a very few instances, we found a critical task that was not supported by technology in a stand-alone application. Chapter VIII discusses future work, and along with the design document for the software that is included as an appendix, that chapter serves as a template for follow-on work.

Endnotes

1. Marine Aviation Weapons and Tactics Squadron One. 2004. *Forward air controller (airborne) handbook*.

2. Commandant of the Marine Corps. 2004. *MCO 3500.48A Aviation training and readiness manual, AH-1*.

3. Commandant of the Marine Corps. 2004. *MCO 3500.49A Aviation training and readiness manual, UH-1*.

4. Expeditionary Warfare Training Group. 2006. *EWTGPAC record of training JFAC(A) 4-06.*

5. Commandant of the Marine Corps. 2004. *MCO 3500.48A Aviation training and readiness manual, AH-1*.

6. Commandant of the Marine Corps. 2004. *MCO 3500.49A Aviation training and readiness manual, UH-1*.

7. Ogus, L.R., Jancaitis, J.R., and Hauck, R.E. 1981. Wargame-based training systems. *1981 Winter Simulation Conference Proceedings*. 667-677.

8. Nyland, W.L. 2002. *Marine aviation campaign plan 2002*.

THIS PAGE INTENTIONALLY LEFT BLANK

II. TASK ANALYSIS

A. FOUNDATION

1. Choosing the Path

One method of developing a trainer is to take the initial training concept and then start assigning the technology that is in vogue. It happens often enough; consider the glut of training systems using head-mounted displays that emerged after that technology became commercially available. When some new gadget allows a more full immersion into a virtual environment, be it by stimulating the visual, aural, or tactile senses, the temptation is great to leverage the feature to build a new application for training.

Taking this path skips over the median step that asks whether the chosen technology might be suited best for the particular skill set being trained. It lacks an analysis to determine which training tasks need to be represented with high fidelity and which may be abstracted. No formal procedure exists for doing this, and it is no small surprise that application development often proceeds through immediate implementation of available technology. We think that this path is less than optimal. Little training occurs when attention is not given to matching goals to tasks. Starting development from a guess of the proper implementation will produce a superior trainer only by pure coincidence.

Perhaps a more intelligent approach is to nail down first what precisely is the goal of your particular training application. By mapping goals, we create a framework within which it is possible to select those critical to training. Keeping those goals in focus during trainer development ensures that the end product remains true to its purpose.

Human abilities in task performance have a stable variance. Perception and response time to stimuli, as well as motor skill activation, remain relatively constant. These things are built into our genome, and they won't be changing anytime soon. Technology evolves much faster, however. This is the main point made by Darken; technology advances more quickly than innate human learning skills.[1] To design a trainer around technology is to chase a moving target. Given that the typical length of application development can be measured in years, technology will advance even before the trainer is built. If trainer design is based on implementing a fantastic new technology instead of founded on human learning principles, then it will often be the case that logical choices made before the development cycle will make little sense afterwards.

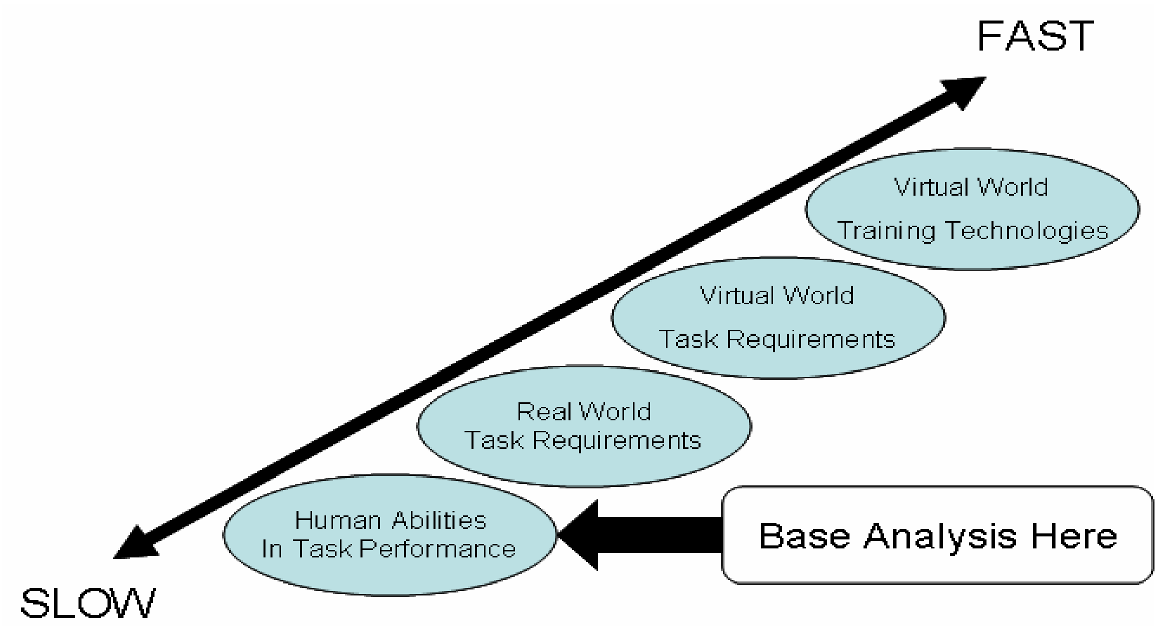


Figure 1. A Solid Foundation[2]

Goals deemed critical are likely to remain so throughout the life of the trainer development cycle, and more importantly, through the near future as technology allows a wider array of implementations. Those goals that currently cannot be trained using existing technology should be documented; this allows later incorporation into training once the applicable technology becomes available and affordable.

2. Identifying Skills

We wanted to know precisely what skills were needed to conduct a FAC(A) mission in order to begin looking for a suitable trainer. A flight simulator would not suffice; we had no intention of teaching aerobatics or even basic stick-and-rudder skills, and in any case a pilot is well-versed in flying the aircraft long before reaching the FAC(A) stage of training. Nor did we want to recreate a virtual representation of the entire mission; certain mundane tasks such as walking out to the flight line and strapping into the aircraft, while necessary, are hardly challenging and contribute nothing toward learning the mission at hand. The key was to find out what was special about doing FAC(A); if the job required proficiency in certain skills, then those were the ones on which we wanted to concentrate.

To identify those skills, we needed to understand how and when the pilots made decisions; this would include knowing what information they required in order to do their job. It required mapping in detail the thought processes and resulting communications that occurred during each phase of a sortie, starting in planning, continuing through execution, and finishing with the flight debrief. In the simplest terms, we wanted to find out what goes on inside a pilot's head when he conducts FAC(A).

Mission success is dependant on making correct decisions. What we sought was a training system designed to allow exercise of those decisions. It would need to force the trainee to use the discrete set of skills that are critical to getting the job done, whatever those skills turned out to be. A dissection of the task was in order.

Having done Forward Air Control before starting this project, as well as having interviewed more senior pilots, we knew the general flow of a sortie. We could describe what took place in the cockpit, but not necessarily *how* it was done, nor could the pilots that we interviewed. Particularly when the task is complex, it is not enough only to observe behavior. To distill the critical skills from the trivial, we had to map out and analyze the decisions made during the course of a typical mission.

B. METHOD

1. GOMS

Card, Moran, and Newell gave us the GOMS process for dissecting a complex task.[3] The GOMS acronym stands for Goals, Operators, Methods, and Selection; applying these labels to individual steps taken in executing a task allows scrutiny of those steps. The idea is that by mapping individual actions to the decisions that prompt them, one should be able to identify the critical aspects of a job. This method of organization can be used for various purposes, from discovering ways to improve the efficiency of a task to revealing essential data needed to perform it. For our purposes, we wanted to know what information we needed to make available within the trainer so that a pilot would have the same resources as he does in the cockpit.

Knowing, for example, that three potential means of marking a target include laser, white phosphorous, and illumination round, implies that the FAC(A) must make a decision as to which mark to use. Exploring these choices reveals that the decision is based, among other variables, on how the environment will affect the marking element. Humidity affects laser beam propagation, and strong winds quickly can spread the smoke of a white phosphorous round to the point that it is ineffective. An illumination round makes an excellent mark, but is not as precise as a laser designation. Mapping the task decisions and the information required to make them reveals that a proposed trainer needs to provide to the pilot certain meteorological data.

The hierarchy of the basic GOMS model is simple. A task is composed of methods that are used to achieve specific goals. At the lowest level, methods are composed of operators. Operators are discrete steps that a user performs, and each operator may be labeled with a precise length of execution. Sometimes, as in the prior example of marking a target, a goal may be achieved by more than one method. In that case, selection rules are used to determine which method takes place.

Operators are classified as perceptual, cognitive, or motor acts. An example of a perceptual operator is seeing a target. Enacting a cognitive operator would be retrieving knowledge about the appropriateness of the various marking methods. In the case of using a laser for marking, a motor operator is used to label the action of pressing the laser designation button in an aircraft.

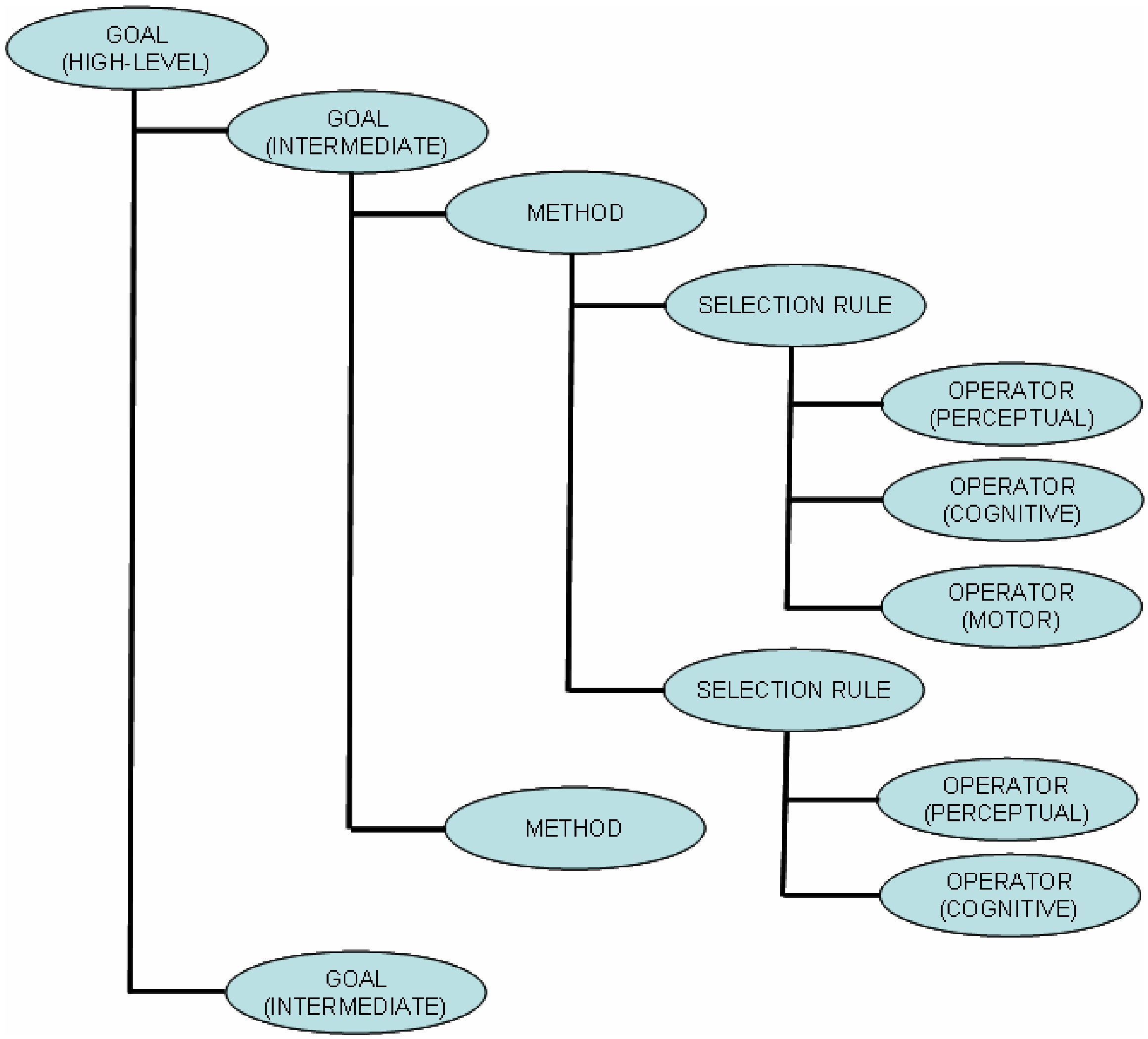


Figure 2. GOMS Structure

Although from the original model of GOMS, there followed several general variations, we chose to use the one introduced by Card et al, as it allows for a pseudo-code format and provides a guide for how to structure selection rules. Commonly know as 'CMN-GOMS,' it foregoes the precision required by more complicated versions and can be tailored for different scenarios.

A prime benefit of using the GOMS model is performance modeling. Individual tasks are given completion times and precedence requirements. Looking at a series of tasks chained together identifies idle periods, which may indicate avoidable gaps in execution. This implies its use as a tool to increase efficiency by rearranging the order in which tasks are completed, either by an individual or the group as a whole.

Our use of the GOMS architecture was for task description only. Although efficiency of the FAC(A) job is certainly desirable, that remains within the scope of other research. We aimed to dissect the evolution of a FAC(A) sortie solely to pinpoint critical learning points during the mission.

2. Mitigation of Drawbacks

There are some drawbacks in using GOMS to describe Forward Air Control. The model does not account for learning during a sortie, and does not accommodate pilot errors. Fatigue is not modeled with CMN-GOMS either; although the effects of fatigue are known to degrade human performance, the model we selected assumes pilot ability remains maximal throughout the flight.

The duration of a FAC(A) training sortie is approximately two hours. Fatigue does set in for longer flights, but based on our experience we believed that it does not dramatically affect decision-making for the short mission under inspection. The trainer we sought would be aimed at those who had not yet performed FAC(A); as such, it would be a preparation tool for starting the short-duration flights. Had our purpose been to find or develop a system that permitted practice for seasoned pilots, we would need to use a more advanced mapping technique, taking into account the effects on performance associated with loss of attention. The benefits of being able to map the mission with elementary methods of perception, cognition, and motor functions appeared to mitigate the aforementioned shortcomings.

C. CONCLUSIONS

1. CTA Scope

Our cognitive task analysis (CTA) followed the execution of a FAC(A) mission beginning with planning and briefing, through execution, and ending with the flight debrief. The analysis is included in its entirety within Appendix A. While not a complete coverage of every aspect of forward air control, it structures the flow of a pilot arriving on station, taking terminal control, building a nine-line attack order, and directing CAS aircraft in its execution. We did not model decisions made when dealing with aircraft emergencies, nor those made in response to taking hostile fire.

2. Critical Skill Identification Process

Taking our cue from the method used in the fleet to document pilot performance, we scoured the CTA for individual tasks and sub-goals that matched evaluation items on the FAC(A) Aviation Training Forms (ATF). An example of one of these forms is shown in Figure 3. Pilots are graded according to a scale that ranges from above average to unsatisfactory. Flight instructors use a fair bit of subjectivity when assessing skill, and for this reason the ATF contains two sections – one for marking letter grades, and another for comments. The comments section can take up several pages, and it was here that we found the most help in determining our critical skills.

Although we were not privy to actual ATFs, we had help from senior flight instructors in data mining for leading causes of incomplete flights. Their consolidated list was a distillation of the comment sections from a library of FAC(A) syllabus ATFs written from 1997 to 2004. Examples included low awareness of unit locations, not maintaining a high operational tempo, choosing poor terrain features for CAS talk-ons, and not ensuring suppression of enemy air defense during CAS attacks.

For some of these skills, it was easy to identify the matching entry on the CTA. For example, our analysis contained a cognitive operator labeled "Choose prominent terrain near target likely to be visible from support aircraft viewpoint." This precisely matched a target skill from the instructor database. For others that were more abstract in nature, such as maintaining a high tempo in the terminal area, we had no exact matches. Implementing a way to train this type of skill would be difficult if it could not be identified by component operators.

Although there were few such skills that eluded a one-to-one match on the CTA, they did pose a problem; we could not claim to meet the needs of the fleet with a trainer for FAC(A) if it did not address a primary training deficiency. Our solution came in the form of a vehicle for after action review (AAR). Specifically in the case of evaluating skills like "maintaining high operational tempo," the AAR provides a means for a flight instructor to judge performance as done during the actual task. Further detail on this method of grading performance and the factors driving our decision to use it is given in Chapter IV (Iterative Development).

FORWARD AIR CONTROL(AIRBORNE) - 343

FAC-343 2.0 C, R 2 AH-1W FAC(A)I/PUI

GOAL: OS - Introduce supporting arms consolidation.

REQUIREMENT: AH-1W with operable LDRS. Pilot will brief a FAC(A) game plan.
(1) Discuss fire support planning documents (target list worksheet, scheduling worksheet) weapon-to-target match.
(2) Review integration of multiple supporting arms assets into objective area mechanics and SEAD procedures. PUI will coordinate SEAD in support of FW target engagement.
(3) Conduct a minimum of 4 FW controls.

EXTERNAL SYLLABUS SUPPORT: 2 FW CAS aircraft with ordnance, 1 indirect fire asset OR 1 section of RW aircraft separate from flight, live fire range and LASER safe range.

1. GENERAL:	**UN**	**BA**	**AV**	**AA**	**DND**
A. BRIEF	☐	☐	☐	☐	☐
B. MISSION PLANNING	☐	☐	☐	☐	☐
C. CHECKLISTS	☐	☐	☐	☐	☐
D. COMMUNICATIONS	☐	☐	☐	☐	☐
E. BASIC AIRWORK	☐	☐	☐	☐	☐
F. SITUATIONAL AWARENESS	☐	☐	☐	☐	☐
G. HEADWORK	☐	☐	☐	☐	☐
H. EMERGENCY PROCEDURES	☐	☐	☐	☐	☐
I. AIRCREW COORDINATION	☐	☐	☐	☐	☐
J. DISCUSSION ITEMS	☐	☐	☐	☐	☐

2. REVIEW:	**UN**	**BA**	**AV**	**AA**	**DND**
A. 9-LINES	☐	☐	☐	☐	☐
B. CONTROL OF FW CAS	☐	☐	☐	☐	☐
B. CALL FOR FIRE	☐	☐	☐	☐	☐
C. TALK-ON TECHNIQUE	☐	☐	☐	☐	☐
D. SEAD	☐	☐	☐	☐	☐
E. FSCM	☐	☐	☐	☐	☐
F. BDA	☐	☐	☐	☐	☐
G. BHO	☐	☐	☐	☐	☐
H. TACTICS	☐	☐	☐	☐	☐

3. TOTALS:	**UN**	**BA**	**AV**	**AA**	**DND**
	☐	☐	☐	☐	☐

4. OVERALL FLIGHT:	**COMPLETE**	**INCOMPLETE**	**UNSAT**
	☐	☐	☐

5. INSTRUCTOR COMMENTS:

IP ________________ PUI ____________
DATE ________________ DURATION __________

Figure 3. Aviation Training Form Example

3. Takeaways

a. Duplicated Skills

A primary take-away from the CTA was that there are many critical skills involved in FAC(A), but many of them are duplicated in other mission types. Some examples include proficiency with target classification, range estimation, and recall of threat capabilities. We dismissed practicing such skills as training objectives whenever they were duplicated as a priority goal earlier in the flight syllabus (Commandant of the Marine Corps, 2003). In squadrons where pilots receive initial training in their Fleet aircraft, several flights concentrate on basic conventional weapons delivery, and require facility with judging range and azimuth to targets, in addition to testing knowledge of adversary weapon systems.

Skills such as the ability to navigate through terrain and maintain adequate aircraft separation from it are important, but they too are developed on non-FAC(A) sorties. By the time a pilot reaches the FAC(A) syllabus, he is proficient in these skills and would not see a challenge in executing them within a trainer.

b. Overly Detailed Tasks

We judged the ability to operate aircraft equipment as critical but not desirable in a training system of this type. Proficiency in manipulating dials and switches enables mission success; however, forcing a pilot to do this in a virtual environment is not in keeping with our vision of a battlefield management trainer. We wanted a higher level of abstraction than that; our emphasis was on allowing the trainee to make things happen without having to go through the minutiae of hitting every toggle required in the actual aircraft.

c. Target Skills

Accurately perceiving the battle space, organizing assets within it, and using those assets to create effective attack packages became our high-level goals. “Puzzle-solving” is the overarching term we used to describe the process of building a solution to FAC(A) problems. Effectively directing assets around a terminal area in order to focus fire on the enemy involves knowing who to contact and when to do it, often maintaining multiple conversations on different radio frequencies. Sometimes it entails vectoring friendly aircraft under the high trajectories of artillery rounds, ensuring the suppression of enemy air defense so that attacking friendly aircraft do not take fire. Knowing the location of a moving forward line of troops is also crucial to success. Furthermore, all this must be done while adapting to a changing battlefield.

4. A Concepts and Procedures Trainer

Having identified the set of skills for the proposed FAC(A) trainer, it became clear that we were looking for a conceptual knowledge trainer, with sub goals of procedural knowledge training. The mission of FAC(A) is an exercise in maintaining maximal situational awareness in terms of knowing unit locations and capabilities, and recognizing opportunities for exploitation of targets. Procedural knowledge training would be key to the process because much of the communication flow during a mission follows standard formats. Knowing what to say is procedural; knowing when to say it is conceptual.

Our goals for the trainer were set. We wanted to enable a user to accomplish seven major tasks, and those tasks became the checklist against which we measured all potential systems. The goals were as follow:

1. Prepare a solid plan of execution
    a. By understanding the Ground Combat Element commander's intent
    b. By knowing the number and types of assets available for tasking
    c. By modifying a chart map and preparing attack packages prior to mission execution
2. Comprehend the battle space
    a. By seeing the Fire Support Coordination Measures, attack packages, and enemy threat ranges
    b. By examining the units on the battlefield
    c. By visualizing the battlefield from other points of view

3. Organize the battle space
    a. By creating and sending 9-lines and fire missions
    b. By maneuvering one’s own aircraft to strategic positions
    c. By specifying special attack parameters
4. Understand how and when reports get processed and by which agency
    a. By coordinating attack plans with the Air Officer
    b. By sending Battle Damage Assessments and Battle Handovers
5. Utilize combined arms for maximum effectiveness
    a. By ensuring Suppression of Enemy Air Defenses when required
    b. By using marking capabilities of supporting arms
6. Effectively use the Talk-on technique
    a. By selecting prominent terrain features
    b. By describing chosen terrain features to CAS aircraft
7. Maintain the operational tempo
    a. By sequencing multiple Close Air Support sorties
    b. By maneuvering Close Air Support through the terminal area

With the recipe for training the desired skill set in hand, we began to look at existing systems that focused on the control of air and ground assets in the prosecution of ground targets. From established multi-user facilities to trade shows and training conventions, we sampled available systems and compared them against the shopping list. Chapter III documents our findings.

Endnotes

1. Darken, R.P., and Cockayne, W. 2003. The application of human ability requirements to virtual environment interface design and evaluation. In *Handbook of task analysis for human-computer interaction*, ed. Dan Diaper and Neville Stanton, 401-421. Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

2. Ibid.

3. Card, S., Moran, T.P., and Newell, A. 1983. *The psychology of human computer interaction*. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

III. COMPARISON OF EXISTING SYSTEMS

A. INTRODUCTION

As stated in Chapter I, one of the motivations for attempting to develop this trainer was to provide FAC(A) syllabus students another resource to better prepare themselves for the four flights in the aircraft. Given the fact that resources such as time, money, and manpower are often limitations that Marine Corps squadrons must attempt to overcome in order to obtain training for its personnel, the determination was made that a lightweight/deployable, open-source solution should be the focus of research. Additionally, the proposed trainer should not require any external resources to operate, such as instructors or projection systems. Based on past fleet experience, it was assumed that such a trainer was currently not in existence and readily accessible to young aspiring FAC(A) pilots; however, prior to designing a trainer that would satisfy the critical FAC(A) tasks as determined by the CTA described in the previous chapter, it was necessary to review systems currently in existence that target similar task training in order to see if our assumption was founded. This research was also critical in aiding the formulation of ideas about how software engineers and other organizations in the past had successfully mapped technology to the task of controlling fires in a virtual environment.

B. CURRENT TRAINING SYSTEMS

1. Call for Fire Trainer (CFFT)

The Call for Fire Trainer is an immersive training system which utilizes 3-D technology to create virtual battlefields in which basic through advanced-level indirect fire call-for-fire missions are supported, including Close Air Support (CAS), Naval Gunfire, and Mortars.[1] The CFFT was designed by Fidelity Technologies Corporation to replace their Guard Unit Armory Device Full-Crew Interactive Simulation Trainer (GUARDFIST II), and it has been marketed as the only U.S. Army-approved system to train call-for-fire procedures. The CFFT is also in use by the Marine Corps at the Army Proving Grounds in Yuma, Arizona. We had the opportunity to witness this system in action during a Fall 2005 training session for forward air controllers in support of a biannual Marine Aviation Weapons and Tactics Squadron One (MAWTS-1) Weapons and Tactics Instructor (WTI) course.

For the purposes of this thesis, the CFFT supports the majority of the critical tasks determined for the proposed system; however, an instructor is required to operate the system, there is a large cost associated with procuring this system, and the CFFT footprint is larger than envisioned for this trainer. The optimal footprint for Cleared Hot is an application that can be distributed on a CD-ROM and run on a laptop or desktop.

Figure 4. CFFT with EVS option (From: Ref. 2)

The Enhanced Visual System (1280 x 1024 projection system) is an available modular addition which upgrades the CFFT's capability for Types 2 and 3 CAS training to Type 1.2 Without going into detail, the leap from Types 2 and 3 CAS to Type 1 can not be made without assurance that the forward air controller can visually acquire the attacking aircraft and the target under attack prior to granting clearance to fire. A trainer capable of simulating all three types of CAS is desirable because in the real world the tactical situation will dictate which type of terminal attack control is utilized. Since the FAC(A) will be given the latitude to determine which types of terminal attack control best accomplish the mission, it is necessary to train and be well-versed in the execution of all three types of CAS; however, the gains achieved in "training like we fight" are undermined here with the substantial increase in footprint size of the CFFT with the EVS option.[3]

2. Indirect Fire Forward Air Control Trainer (I-FACT)

The Indirect Fire Forward Air Control Trainer is a commercial-off-the-shelf (COTS) product designed by FATS Incorporated to support the unique training needs of Joint Close Air Support Terminal Attack Control training.[4] It

> ...has two training modes, known as “Virtual” and “Pilot-in–the-loop” training. Using the Virtual training mode, forward air controllers can conduct fixed and rotary wing close air support missions, close air support using bomber aircraft, and AC130 gunship close air support. These elements, coupled with the supported indirect fire assets, allow trainees to conduct procedural training, close air support planning, coordination and de-confliction of aircraft, suppression of enemy air defense, marking of targets, and the simultaneous attack of targets using both fixed and rotary wing aircraft. For more advanced close air support training the Pilot-in-the-loop mode allows an individual to pilot the virtual aircraft acting as fixed or rotary wing aircraft. In this mode, ground controllers coordinate directly with the pilot and work to talk the pilot on to ground-based targets just as they would in the real world.[5]

Figure 5. I-FACT (From: Ref. 5)

Similar in functionality to the CFFT, the I-FACT is an extraordinarily versatile and complete training solution; however, once again, one must contend with the cost associated with the system and the instructor requirement for operation of the trainer.

3. Multi-purpose Supporting Arms Trainer (MSAT)

The effort to develop the Multi-purpose Supporting Arms Trainer was created in 1997 by a Mission Needs Statement submitted by the Expeditionary Warfare Training Group Atlantic (EWTGLANT), which has partial responsibility for providing training to Marine Corps, Navy, Army, Air Force, and Special Operations personnel to include

Forward Observers/Spotters, Forward Air Controllers, and Terminal Attack Controllers.[6] This enormous responsibility is shared with the Expeditionary Warfare Training Group Pacific (EWTGPAC). The two training groups were in need of a replacement for their obsolete and unsupportable forward air controller and forward observer training device.

According to a brief written by John Bilbruck, Program Manager for MSAT at the Naval Air Warfare Center Training Systems Division Orlando, the supporting arms trainer is scheduled for Phase 2 acceptance by EWTGLANT this summer.[7] The primary functionality delivered in Phase 1 last year provided for voice automated ground call for fire and limited CAS capability with “man in the loop” controlled Tactical Air (TACAIR) assets via the Marine Corps Deployable Virtual Training Environment (DVTE—See Chapter VIII for further detail). Complete voice automated CAS functionality with Joint Semi-Automated Forces (JSAF) driven scenario control and After Action Review (AAR) are the highlights of planned Phase 2 feature enhancements. MSAT seems poised to greatly enhance TACP training for all future forward air controllers (ground and airborne), given one has the opportunity to attend an EWTG ground school. Nevertheless, the size of this system will limit its application in many environments.

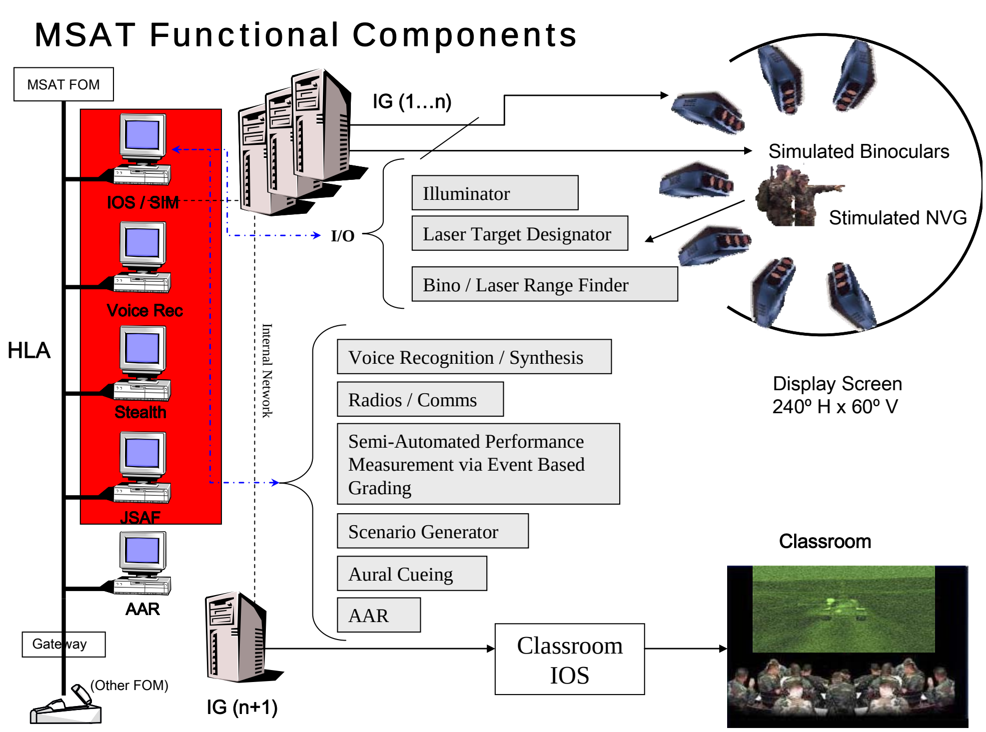


Figure 6. MSAT Functional Components (From: Ref. 7)

4. Forward Air Controller Training Simulator (FACSIM)

The Forward Air Controller Training Simulator, developed by TNO and utilized by the Dutch Air Ground Operations School (AGOS), was on display at the Interservice/Industry Training, Simulation and Education Conference 2005 in Orlando, Florida. The impetus for the design of FACSIM closely matches the motivation for *Cleared Hot*. FACSIM

> ...provides the solution to the training problems: it fills the gap between classical classroom training and live training. The system provides a simulated environment in which the trainees can fully build-up skills, thus preparing for the real work. Rather than a simple procedure trainer, the system is an integral task trainer. It creates the required training environment in which the trainees can experience all the responsibilities involved in the controller task, thus learning to apply the correct skills at the right time in a confident and effective manner. The training simulator thus is an indispensable tool that yields: higher standards for initial training, improved proficiency of combat ready controllers, and optimal use of live training controls.[8]

The key functionality inherent in FACSIM is indicative of the features required for this thesis, but FACSIM incorporates computer stations for the CAS pilot, the Laser Target Designation Officer (LTDO), and the supervising instructor who is responsible for overall execution of the training event. Due to these manpower requirements, along with the compulsory cost of the system, FACSIM is rejected as a viable solution to the research question at hand.

5. Synthetic Teammates for Realtime, Anywhere Training and Assessment (STRATA)

CHI Systems Incorporated is the developer of Synthetic Teammates for Realtime, Anywhere Training and Assessment. STRATA is currently being employed as part of DARWARS, which is a Defense Advanced Research Projects Agency (DARPA) funded project to accelerate the development and deployment of the next generation of experimental training systems.[9]

The technology employed in this system is suitable to satisfy the task requirements as set forth in Chapter II. Additionally, the goals and objectives of STRATA, as set forth in the following quote, very closely resemble the motivation for this thesis:

> Through a novel integration of speech-interactive synthetic teammates, intelligent tutoring, and scenario-based training, Synthetic Teammates for Realtime Anywhere Training and Assessment, or STRATA, overcomes conventional training limitations by providing fully deployable, effective, and engaging training that offers on-demand practice for individuals and teams with or without instructors – and – with or without the team.[10]

Other desirable features of STRATA include: an application structure that is interoperable with the High Level Architecture (HLA) utilized in the Marine Corps' DVTE, after-action review, application of airspace control measures for procedural control of aircraft routing, and its emphasis on headwork and decision making over airmanship.

STRATA is capable of providing synthetic teammates for the CAS lead aircraft, wingman, and FAC. Although the capability exists for the user to play the role as FAC with artificially intelligent CAS, the system's main objective appears to be CAS training, not FAC or FAC(A) training. Nevertheless, on the surface, the implementation of speech interaction between the FAC and CAS appears to be an excellent match for one of our seven trainer goals: allowing practice of the "talk-on" for CAS aircraft. However, upon further research, a synthetic FAC communicating with the user is not the same problem as its converse. The converse presents different challenges that will be detailed in Chapter IV. Consequently, an integration of the CHI system with *Cleared Hot* was later attempted, but it failed due to the fact that one of the objectives of *Cleared Hot* is to be completely open source and there were licensing issues associated with STRATA.

6. Forward Observer Personal Computer Simulator (FOPCSIM) 2

The Forward Observer Personal Computer Simulator 2 was developed in 2005 by McDonough and Strom at the Naval Postgraduate School in order to increase training opportunities for artillery forward observers in the Marine Corps. FOPCSIM 2 builds upon FOPCSIM 1, a forward observer simulator designed in 2002 by Brannon and Villandre.[11] Although FOPCSIM 2 does not support the task of Suppression of Enemy Air Defenses (SEAD) and therefore CAS is not incorporated in the trainer, the basic call for fire task executed by forward observers is itself a critical task for the FAC(A). Consequently, this thesis would need to utilize the work done by McDonough and Strom as a foundation for expansion.

FOPCSIM 2 has already enjoyed widespread success within the Marine Corps as evidenced by its recent use at The Basic School, Expeditionary Warfare School, and the School of Infantry – East.[12] This thesis contends that the success of FOPCSIM 2 is largely due to the fact that it "was developed at no cost, is freely available, takes advantages of modern 3D graphics, eliminates costly contractor support, and will run on laptops in support of deploying units."[13]

FOPCSIM 2 is built upon Delta3D, an open source gaming and simulation engine developed by the Modeling, Virtual Environments and Simulation (MOVES) Institute at the Naval Postgraduate School. Delta3D is released under the GNU Lesser General Public License (LGPL), which makes it freely distributable. See Chapter V for more detail.

Figure 7. FOPCSIM 2 (From: http://www.delta3d.org retrieved on May 3, 2006)

C. CONCLUSION

Our review of existing trainers found many excellent candidates; each addressed several of our training goals. Only FOPCSIM 2 offered an open-source solution, however, and it did not allow control of CAS aircraft. The remaining systems that did support CAS integration into battle space control lacked the capability to train a pilot without the need for external support. In many cases, these systems also presented a sizeable footprint, lessening their portability.

We decided to develop our own trainer. FOPCSIM 2 had shown us that an open-source model can be successful; we planned to base our own system on it. Having identified the skill set critical to conducting FAC(A), we began translating our ideas for training it into a design document. The next chapter tells the story of that process.

Endnotes

1. Fidelity Technologies Corporation. *Call for Fire Trainer (CFFT)*. (2005). Retrieved May 3, 2006, from http://www.fidelitytech.com/cfft.htm.

2. Fidelity Technologies Corporation. *Enhanced Visual System (EVS).* (2005). Retrieved May 3, 2006, from http://www.fidelitytech.com/EVS.htm.

3. Joint Publication 3-09.3 (2005). *Joint Tactics, Techniques, and Procedures for Close Air Support (CAS), V-14*.

4. Haley, T. T. (n.d.). *DRAFT Indirect Fire Trainer (IFT)/Indirect Fire-Forward Air Control Trainer (I-FACT) Capabilities Document*, 5.

5. Fats Virtual Training Solutions Company. *Indirect Fire Forward Air Control Trainer*. (2006). Retrieved May 3, 2006, from http://www.fatsinc.com/I-FACT.htm.

6. Naval Air Warfare Center Training Systems Division (NAWCTSD) Orlando, FL. (2003). *Multipurpose Supporting Arms Trainer (MSAT) Instructional Performance Requirements Document (IPRD)*, 7.

7. Bilbruck, J. (n.d). *Multi-purpose Supporting Arms Trainer (MSAT) Overview*. [PowerPoint brief].

8. TNO. *Forward Air Controller Training Simulator (FACSIM)*. (n.d.). Retrieved May 3, 2006, from http://www.tno.nl/defensie_en_veiligheid/commandovoering_en_operat/simulatoren/facsim/S050113_FACSIM.pdf.

9. CHI Systems Inc. *Synthetic Teammates for Realtime, Anywhere Training and Assessment (STRATA)*. (n.d.). Retrieved May 3, 2006, from http://strata.chisystems.com.

10. CHI Systems Inc. *Synthetic Teammates for Realtime, Anywhere Training and Assessment (STRATA)*. (2006). Retrieved May 3, 2006, from http://strata.chisystems.com/downloads/STRATA_InfoSheet_02-06.pdf.

11. Brannon, M., & Villandre, D. (2002). *Forward Observer Personal Computer Simulator, Master's Thesis*. Naval Postgraduate School, Monterey, California.

12. McDonough, J. P., & Strom, M. S. (n.d.). *Forward Observer PC Simulator 2 Version 1.0.* [PowerPoint Brief]. Retrieved May 3, 2006, from http://www.nps.navy.mil/cs/Research/vissim/ForwardObserver/main.htm.

13. McDonough, J. P., & Strom, M. S. (2005). *The Forward Observer PC Simulator (FOPCSIM) 2, Master's Thesis*. Naval Postgraduate School, Monterey, California.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. ITERATIVE DEVELOPMENT

A. VISION

Starting some 18 months prior to the publication of this thesis, we had no idea what a FAC(A) trainer would look like, nor did we want to form any ideas that would prematurely influence what technologies would be used in it. We did know, however, that whatever the final design, it would need to be a standalone application. Our vision was to provide a portable product to the target user – the squadron pilot. It needed to work without anyone but the trainee using it, and the system had to be open-source.

We knew what the trainer would *not* look like, however. We would not be building a flight simulator. Executing the mission of FAC(A) involves very little flying. The mission commander's plate is full with the job of organizing the battle space; typically he directs the copilot to maneuver the aircraft. Requiring a user to perform both flight and command duties would be demanding more than a pilot does during the actual task. Executing a virtual mission that differs so much from the real one is not likely to generate positive training.

Realizing that industry developers produce software with larger teams and more time than afforded to us, we started at a disadvantage and could not make many wrong turns. We caught several lucky breaks in this respect; an open-source game engine was in development at the Naval Postgraduate School, the engineers that pioneered it were available to adapt it for use in our trainer, and a funding agency was interested in seeing the results.

B. PROJECT TEAM

1. Rudder Steers

We benefited greatly from the guidance of two individuals that had witnessed the creation of similar military trainers. Our thesis advisors had seen the development of both FOPCSIM and FOPCSIM 2. They knew the dangers of wading into a design process prior to mapping critical tasks to technology, and pulled us back to center whenever we began to drift in that direction.

2. Software Engineers

Without the long hours and industry experience of four software engineers and two graphic artists within the MOVES department of NPS, the project would never have taken off. The coding and graphics team consisted of four programmers and two graphic artists. They listened patiently while we described the FAC(A) mission and presented our designs. We often needed to make a choice between equally ideal means for information display; the engineers offered advice as to which choice adhered more closely to commercial standards. The trainer even owes its name to the engineering team; breaking away from the military tendency towards unexciting acronyms, they suggested the system's title be taken from the command given by a FAC(A) to authorize ordnance release, and thus *Cleared Hot* was christened.

3. Subject Matter Experts

Our role in this project was to determine the critical skills of the mission, to provide some measure of subject-matter expertise, and to produce a design document that laid out how the trainer would operate. Our responsibility to the Naval Postgraduate School was to document the process of how the trainer came to be. We hope this thesis will serve as a template for those who seek to build military trainers based on a foundation of task analysis, in addition to being a blueprint for adding extended functionality to Cleared Hot itself.

Questions often arose during this cycle to which we had no immediate answers. Although we had served as Forward Air Controllers in years past, changes in Joint Doctrine had made some of our experiences outdated. One value we brought to the project was a connection to the fleet. When the need for details outside our knowledge came up, we could provide them with a few phone calls to friends and contacts.

We tested versions of the software as it evolved, giving feedback on the fidelity of the agent flight models, behavior of trainer tools, and the presentation of data displays. A daily interaction with the engineers meant that conflicts in design could be resolved quickly, and greatly hastened the development time. In addition, we were able to provide to the graphic artists photo samples of aircraft and cockpits for their incorporation into model textures.

4. Funding

Virtual Technologies and Environments (VIRTE), a program under the purview of the Office of Naval Research, has the responsibility to research and develop a family of training simulators for Navy and Marine Corps expeditionary warfare. Program representatives became interested in the *Cleared Hot* project after a briefing in September 2004, and desired to see it developed as part of a networked system of trainers. Funding from VIRTE made it possible to begin work; although our vision was for a stand-alone application, we were able create much of the trainer interface while building a version that was High Level Architecture (HLA) compliant.

C. PROCESS

1. Cross-Training

As novice as we were about the coding process, the engineering team was just as unfamiliar with the mission of FAC(A). Before we could begin discussions of matching training goals to technology, the people doing the implementation wanted to understand how it would all flow together. As SMEs, we needed education on what was possible to do with the Delta 3D engine as well. It was a learning process for all of us; this stage of development was about cross-training.

We spent several weeks talking about typical FAC(A) missions, answering questions on where pilots get information, with whom a pilot speaks, and what were the priorities of the mission. Many of our identified critical tasks seemed nonsensical until we explained how a sortie can fall apart if the skills needed to accomplish them are weak. One such example was the task of specifying special attack parameters in a nine-line attack order. Until we walked through a scenario of a simultaneous, sectored attack by two sections of CAS, it was not clear why a trainee would need to be proficient at giving those aircraft limits to their usable airspace.

In addition, we learned how presumably simple ideas were limited by system memory and therefore could not be implemented precisely as desired. Having a configurable digital map was possible, but it could not cover the same amount of area as a standard 1:50,000 scale tactical chart. Available military map resolutions produced loading times that made using the full area impractical without specialized hardware.

2. Mapping Task to Technology

a. Identifying Potential Implementations

For the initial stage of development, we relied heavily on the experience of the engineering team. We began with our list of critical skills, a sample of which is shown in Figure 8. Having seen the design of many commercially successful Real-Time Strategy (RTS) games, and researched several during the early stages of development, we knew some of the implementation possibilities, but making those functions work in Delta 3D had not been done before. Given our limited knowledge of what was possible to do in software, we pointed out examples from off-the-shelf products, and the engineers gave feedback on the viability of the design.

Running down a modified CTA, which at this point included two new columns for noting whether technology supports training the skill and how it could be done, we began to whittle down the roster of training goals. It appeared that not every skill set we deemed critical could be trained with fidelity to the actual task. Where this was the case, we decided to forego a poor implementation in favor of none at all.

TRANFER ITEMS	CRITICAL TRANSFER ITEM	CURRENT TECHNOLOGY FACILITATES	HOW FACILITATED (TECHNOLOGY)
12.2.3.2 METHOD: Conduct attack			
12.2.3.2.1 METHOD: Aurally acquire support aircraft			
12.2.3.2.1.1 OPERATOR(P): Hear IP inbound call ((callsign) (IP name) inbound)			
12.2.3.2.1.2 OPERATOR(P): Scan target area			
12.2.3.2.1.3 OPERATOR(C): Choose prominent terrain near target likely to be visible from support aircraft viewpoint	X	YES	Delta3D engine terrain resolution sufficient for airborne view discrimination
12.2.3.2.2 METHOD: Visually acquire support aircraft			
12.2.3.2.2.1 OPERATOR(P): See Initial Point on map	X	YES	2D map displays common
12.2.3.2.2.2 OPERATOR(P): See your location on map	X	YES	Icon representation on 2D 'overhead view' map
12.2.3.2.2.3 OPERATOR(C): Determine azimuth from which support aircraft is likely to appear	X**	YES	Moving 'blip' representation of support aircraft on 2D 'overhead view' map
12.2.3.2.2.4 METHOD (Does not preclude continuation of follow-on methods): Visually scan appropriate azimuth for support aircraft			
12.2.3.2.2.5 SELECTION: If support aircraft is in visual range:			
12.2.3.2.2.5.1 OPERATOR(M): Report Visual			
12.2.3.2.2.5.2 METHOD (Does not preclude follow-on methods): Provide talk-on			
12.2.3.2.2.5.2.1 METHOD: Use visual 'funnel' for support aircraft talk-on			
12.2.3.2.2.5.2.1.1 OPERATOR (M): Query if support aircraft sees largest feature in target area (Do you see the ridgeline running north-south?)	X	NOT WELL	Voice recognition not sufficiently advanced; potential use with limited vocabularies
12.2.3.2.2.6 SELECTION: If support aircraft is not in visual range:			
12.2.3.2.6.1 OPERATOR(M): Report Continue			
12.2.3.2.6.2 METHOD: Use METHOD: Visually scan appropriate azimuth for support aircraft			

Figure 8. Sample of critical skill to technology mapping

b. Avoiding Negative Training – Case of the CAS Talk-On

The “Talk-on” is a term used to describe one of the conversations between a Forward Air Controller and a pilot conducting CAS. It consists of plain-language descriptors that serve the purpose of focusing the CAS pilot’s eyes on the target area. Offered by fleet instructors as one of the main weaknesses of syllabus pilots, the talk-on requires one to choose prominent terrain and man-made features that would be distinct to a pilot at altitude. Figure 9 shows an example of this.

The difficulty in implementation stemmed from our determination to make the trainer a standalone application; i.e., no external support would be required while a pilot used the system. While all of the previously reviewed systems capable of exercising the talk-on use this kind of support, ours would not. It meant that a software agent playing the part of a CAS aircraft would need to understand the meaning of the trainee’s descriptions. This breaks down into two problems, voice recognition and semantic reasoning.

To address the first problem, voice recognition technology has become robust in recent years; it was conceivable that we could leverage it to convert a trainee’s verbalizations into text strings. An implied task in doing this was locating an open-source module for voice recognition; however, no such applications existed. At any rate, our budget did not cover licensing fees for even one application, let alone enough to supply all target trainees.

The second problem was the major obstacle. A software agent that is capable of parsing a text string and matching that to terrain features—as would a human being—is an evolution in artificial intelligence that does not yet exist. Consider the first question shown in Figure 9; “Do you see a prominent mountain range running generally northwest to southeast through Panther?” Certainly an agent could reference a database with key words corresponding to certain terrain such as *prominent mountain*, but this would entail generating all such possible matches for each training scenario. It would also be subject to personal definitions of what constitutes prominent terrain. If even one such feature were left out of the database, imagine the frustration of the end user who picks a perfectly legitimate mountain, only to receive a response from the CAS agent such as "No, I do not see it." Knowing that fleet instructors do not have the luxury of excess time to configure software scenarios, we were not eager to go down this path.

FAC(A):	"Proceed to Panther, establish a left hand pattern at base plus eight and above, report established."
CAS:	"Roger, … established."
FAC(A):	"Do you see a prominent mountain range running generally northwest to southeast through Panther?"
CAS:	"Contact."
FAC(A):	"Those are the Chocolate Mountains."
CAS:	"Roger."
FAC(A):	"Look to the northeast of the Chocolate Mountains from Panther for a single large dark mountain. Do you see it?"
CAS:	"Contact."
FAC(A):	"Describe the general shape of the mountain that you see."
CAS:	"Roughly arrow shaped with the point oriented northwest."
FAC(A):	"Correct, that is Blue Mountain. Look at the northwest end of Blue Mountain, the arrow points to a dirt runway. Do you see the runway?"
CAS:	"Contact."
FAC(A):	"What direction is the runway oriented?"
CAS:	"Generally north-south."
FAC(A):	"Correct."
FAC(A):	"Look at the north end of the runway. What do you see?"
CAS:	"I see six trucks in a circle."
FAC(A):	"That is your target. Attack the eastern most truck. Your final attack cone is 210° to 270°, report rolling in with your direction."

Figure 9. Example of the Talk-On

One very novel idea for solving the talk-on problem came from one of the software engineers. The idea was to let the trainee do his own critique of prominent terrain selection. The user would use a non-verbal means for choosing a feature; a mouse-click on any group of pixels in the visible area would generate a coordinate trio. This would constitute the implied question, "Do you see what I have clicked?" Following that action, the user would toggle to the viewpoint of the CAS pilot. If, from that perspective, he could select the same terrain feature (same set of pixels), then it would be deemed an acceptable one.

While this idea became our most promising means for implementing the talk-on, it never came to fruition. Before solidifying plans for a certain aspect of the trainer, we would typically debate potential ways to game the system; i.e., ways that a user could exploit shortcuts and flaws in design. For this particular plan, we discovered that by selecting a large enough area from either perspective, a trainee would be guaranteed a pixel match. As with other potential implementations, if the possibility existed to generate negative training, we tabled the option in favor of design choices that could be made immediately.

Of note for this example is the method by which we decided to leave out the "Talk-on." It highlights the importance of keeping engineers and SMEs in close contact during the development process. Given the requirement to facilitate a "Talk-on," programmers working in solitude would undoubtedly make it happen; in this case the result would allow for exploits of the system that would negate potential gains in training. By maintaining a discussion between all design parties, we were able to quickly discount implementations that could not be perfectly executed.

c. Avoiding Negative Training – Case of the After-Action Review

To find ways of training some skills, we faced a slightly different dilemma. Judging a user's overall performance at FAC(A) is based on both the end state (Did he accomplish the GCE commander's intent?) and the thought process used to arrive there (Were his decisions sound?). It is sometimes the case that good choices result in less than optimal outcomes, and vice versa. Consider that a trainee may arrange a series of air attacks on a column of tanks without conducting reconnaissance to determine if adversary ADA is present. Clearing those attacks in quick succession will result in a very high destruction rate of the enemy forces, and if there was in fact no air defense around to pose a threat to CAS, then the trainee's actions would appear correct. A system that judged only end results would necessarily declare the trainee's performance superior.

On the other hand, imagine a trainee that spent 20 minutes using the simulated aircraft sensors to visually scour the terrain for the presence of threats to CAS. If he finds none, he proceeds to direct the CAS in their release of ordnance on the tanks. The end state is the same for both trainees; the tank column is destroyed. Therefore, what is the impetus to use due diligence in scouting the area prior to the attacks? One could argue that the foolish student will be successful only a percentage of the time, while the careful one will always be correct; however, we were not prepared to design a system that evaluated end states alone. A method of looking at thought processes was required.

Given the variety of ways that any given attack sequence may be set up, only another human being—an expert in the domain—is capable of sifting through someone's actions and declaring the resulting approach to mission accomplishment as either sound or unsound. Did the trainee consider prevailing winds and their effects on the smoke from a marking round? Did he bring in CAS from the direction of the sun, or from abeam it due to the presence of a high threat in its direction? In fact, we began to develop the logic gates that would filter trainee decisions in these matters. When it became clear that this structure would require data not provided by the game, we looked for a simpler solution.

Fortunately, the framework exists to allow an evaluation of a trainee's cognitive process. Within USMC squadrons, a Weapons and Tactics Instructor (WTI) is charged with the education and development of junior pilots. In particular, the WTI conducts a thorough debrief following each sortie. The debrief is the vehicle by which the majority of learning is done; it presents an environment devoid of distractions and multi-tasking (such as in flight). Pilots being debriefed get an opportunity to explain their thought processes for decisions made during the recent sortie, learn alternative–often better–means for mission accomplishment, and ask questions about the larger battle space in which they operate.

We decided to leverage the expertise of the WTI for the trainer. A debriefing mode would be included, allowing the playback, pause, and rewind, in VCR-fashion, of all user actions during a training session. It was not a perfect solution; by doing this we were transgressing on an original design philosophy. *Cleared Hot* was meant to be a standalone application; its selling point was that it needed no external support. Our compromise was to maintain that during conduct of a training session, it still required no instructor support; however, to give full benefit to a trainee, an instructor would be needed for debriefs.

3. Weekly Meetings

a. Moving Forward Into Design

Immediately following the cross-training, when we felt that we understood how the project would evolve, and the engineers were clear on the major aspects of the FAC(A) mission, we began designing in earnest. During weekly meetings, we discussed more skill training methods, gathered requests for information, talked about the application's look and feel, and decided on flow of a representative mission. With respect to flow, one of our first tasks was to decide how the user moved among game states. Chapter V covers this in detail from a technical standpoint; it is mentioned here as a background to introducing a few design techniques we found helpful.

b. Storyboards

Pencil sketches of game flow worked well for presenting our ideas to the engineering team. Early on in development, we had been advised to keep the drawings simple, not to worry about crafting anything in Microsoft® PowerPoint® or CAD software. This turned out to be good advice. Ironically, suggestions for improvement came more freely when the sketches looked like we had not invested a significant amount of time in generating them.

c. The Note Card Method

The game engine with which we worked functioned as a state machine. To provide an example, this meant the application could start in a Mission Briefing state, move to a Sortie Launch state, and then into an Area Reconnaissance state, to name a few. Microsoft® Visio® was a good tool to use in organizing states, and we went through several versions of design before arriving at a solution. Unfortunately, none of the intricate state diagrams we produced passed muster with the engineers; within minutes they were able to find flaws in state flow.

By chance, we discovered a highly successful technique for building the state diagram. Prior to one of the weekly meetings, we were attempting to structure the flow once more, and decided to use simple 8 x 5 inch note cards, on each of which was written the name of a game state. Once we thought we had a solution, we brought the cards to the meeting, laid them out on a conference table, and talked through them. The cards filled most of the available table space. When anyone raised a valid objection to the organization, we invited that person to adjust the cards slightly. This worked very well; people were reaching across the table, moving cards about, and discussing novel ways to get the machine to work. Within a few hours, we had a final version of the state diagram, never needing to use anything but pencil and paper.

4. Design Document

a. Team Co-location

Chapter I discusses the danger of designing a trainer without identifying critical skills. Another pitfall is the act of submitting a requirements or design document to a team of engineers and then walking away from the project. The expectation is that every aspect of the trainer’s operation has been hammered out, and many months later the authors of the document can come back to find functional software that adheres to their original vision. The problem with this method is that the engineers are forced to make design decisions in a void. Like a battle plan, no requirements document survives the first conflict intact. Subject matter experts of the skill set to be trained may have written the design document, but they will need to be consulted again and again as the code is written. Without such liaison, programmers will be forced to make best guesses on design issues, and it may result in a trainer vastly different in functionality than originally planned.

In our case, the subject matter experts and the engineers were collocated for the duration of project development. This was not necessarily by design; we were pursuing a degree at the time but the arrangement worked well. Whenever a design question came up, we could usually provide an answer within hours. That was fortunate because there were literally hundreds of details that we had not covered in the draft requirements document.

b. Scope Definition

Our initial attempt at defining the scope of the trainer was overly ambitious as well. We had a rather naive view of what goes into software development. As mentioned previously, the wish list of trainer features read like a composite of all the best functionality from commercially successful RTS games. We soon came to appreciate the amount of work that goes into developing products of that caliber.

Additionally, as time passed, we better understood the process of human learning. Taking courses in training transfer provided insight into the value of modular training; i.e., taking a particularly difficult task and training it separately, and when it is mastered, allowing the trainee to use the new skill in the larger application. Incidentally, this became the final solution for the talk-on problem previously mentioned. As it was not currently feasible to train the talk-on skill in a standalone application, we decided to pull it into its own module. The idea was that a trainee would practice the talk-on, and after proving mastery of it, move onto the broader exercise of the complete FAC(A) mission. The talk-on module remains on our wish list of features to be implemented once technology can facilitate it.

c. Evolving Design

Appendix D contains the Game Design Document in whole. It is the sum of our decisions regarding the trainer look and feel, how state transitions flow, the way that agents make decisions, and what information is available to the trainee. It represents a process of distillation and adaptation; in parallel to the decision process by which we segregated critical tasks according to technical implementation ability, so the design document underwent similar changes.

That document is also a wish list of sorts. It serves as a map for future development. Not every described feature was adapted in the final version of *Cleared Hot*, although we have made efforts to denote within it when that is the case. Further documentation on that topic follows in Chapter VIII (Future Work).

The Game Design Document became our information clearinghouse. Posted on a shared network drive (Sharepoint), all members of the development team used it to record the results of changes discussed in meetings, and to post new ideas generated independently. While it began as a few notes on game state flow, over time it became the representation of the entire project.

5. Consultations

a. Fleet Instructor Pilots

One of the first research trips we made was to Marine Aircraft Group 39 based at Camp Pendleton, California. It is the home to four rotary-wing squadrons that conduct the FAC(A) mission. The corporate knowledge of instructors in those squadrons was a valuable resource; it was from these men and women that we received our most valuable guidance regarding what the fleet would like to see from our proposed system.

The idea of implementing a scoring system within *Cleared Hot* was met with a unanimous veto. Although player scores give an indication of accomplishment in recreational video games, for the purposes of a military trainer, the instructors felt that concrete scores could be misleading. The problem is that actual sorties are graded somewhat subjectively. There are concrete standards that a pilot may meet such as controlling some number of CAS sections, or destroying some quantity of enemy armor, but these numbers would have little to do with whether the correct cognitive process was being used.

While having a final score was rejected, the instructors did like the idea of reporting metrics. Seeing a report of time between attacks, number of times SEAD was employed, and error rate in passing doctrinal communication formats appeared to be a great idea. Although the final judgment on pilot performance rests with the instructors, this type of data would help them form their decisions.

We asked if it would be desirable to train the mission in a virtual night environment as well. The engineering team had informed us that while it was possible to slap a green tint on the 3D world to simulate a view through night vision devices (NVD), it was not likely we would be able to see heat blooms from environmental objects with any level of fidelity. The response from the fleet was reassuring; they saw the system as a preparation tool for the FAC(A) syllabus. As pilots would be using it prior to conducting the mission during daylight, there was no need for simulated NVD operations. If *Cleared Hot* could offer an opportunity to train in the basics—and do that job well—then we could forego implementing varsity level functionality.

b. MAWTS

The cognizant officers for FAC(A) at MAWTS-1 gave generously of their time in phone consultations and early visits during development as well as for the proof of concept presentation after a working version of *Cleared Hot* was complete. Their ideas for potential training metrics, such as time between aircraft controls, proved invaluable.

In addition, they provided guidance on the design of certain GUI elements such as the stack diagram. Their suggestions after seeing the finished product are captured in Chapter VII (Conclusions).

c. EWTGPAC

Discussions with the war gaming staff at EWTGPAC aboard NAS Coronado revealed current methods used to give FAC(A) syllabus pilots practice in building attack packages during the one-week course in their facility. Learning how these exercises were structured helped us to model the scenario used in the proof of concept release of *Cleared Hot*.

d. Local Talent

The project benefited from consultations with several experienced U.S. Air Force officers attending the Naval Postgraduate School as well. From these individuals we received many new ideas for trainer metrics, but specifically the concept of a tiered system for making the simulation progressively more difficult. An example of this was their suggestion to have CAS aircraft fuel and ordnance states automatically update for beginning trainees, but to force more advanced learners to manually change the data.

e. Focus Groups

The Human Factors and Training Focus Group within the MOVES department of NPS holds a weekly meeting to discuss topics of interest in the fields of HCI and training transfer. The group allowed us to present our trainer designs on two occasions after we had produced initial GUIs for *Cleared Hot*. This proved to be a tremendous help in finding flaws in the interface; with an audience of dozens analyzing the product, we quickly learned where it could be made more intuitive for the end user.

6. User Study – Case of the Unit Communication Interface

a. Problem Statement

Central to the trainer's effectiveness would be the ease in which a user could communicate with agents under his control. We knew an ungainly interface would frustrate trainees. Command selections hidden deep within menus, ambiguous icon functionality, and in the case of text entry fields, a low tolerance for spelling mistakes, all typically result in user rejection of a trainer.

We developed two potential designs for a GUI used to transmit orders to units; the two test designs are shown in Figures 10 and 11. Using Java, we built them for the purpose of conducting a user study; the results would help us design the communications interface for *Cleared Hot*. We compared best industry practices with experimental research; our goal was to minimize the artificial delays in command transmission necessitated by navigation through an interface.

When engaged in a multi-user simulation, passing orders is simple; users employ an analogue of a radio. The trainee experiences no unusual degradation of communication flow. The practice of keying a handset and speaking to another human being during a simulation is consistent with the communication practices in operational environments. There is no need in such cases to design a GUI for transmitting simple commands such as “flank the enemy to the south,” or “take position west of the hill at 12 o’clock.” For our standalone application, however, there would need to some other method.

b. Research and Design of Test GUIs

The commercial video gaming industry annually releases dozens of games in the real time strategy venue that employ GUIs to direct unit actions. De facto standards have arisen with respect to how a user moves units around a simulated battlefield. These practices seemed to be fertile ground from which to distill a set of design principles for our communication interface.

In addition, there exists a large body of research on user interaction with GUIs. Marcus discusses consistency within displays and effective use of color.[1] Publications of industry standard practices, such as by Apple Computer, Incorporated describe useful architectures for GUIs.[2] Endsley, Bolté, and Jones list 50 design principles they believe will improve situational awareness.

Pertinent to a communication interface, one guideline from Endsley et al. particularly stands out. The first is the mandate to explicitly identify missing information. Military radio traffic is characterized by a discrete set of standard phrases. Although plain language occasionally is required to further a message recipient’s understanding, the pattern of communication follows the format of “You, this is me; do this action, at this time in the future, for this purpose." Given a limited set of potential actions, times, and purposes, this message format may be represented in a GUI by "either/or" selections as opposed to blanks that must be filled in with text by the user. If the list of possibilities is finite, Endsley et al. suggest creating an environment that guides the user to the correct set.[3]

In addition, there is no dearth of experimental data from user testing on various popular GUI designs. Whiteside, Jones, Levy, and Wixon applied a holistic approach to experiments using three leading designs. Their results showed that iconic interfaces outperformed menu-based systems.[4] When possible, we want our command and control GUI to avoid using drop-down menus for high-level unit selection. For the specific applications they were testing, Whiteside et al. found users more quickly executed commands when they could see the icons representing those commands at the highest level of interface. Taking the results of this research as a cue, we designed the first test GUI to persistently display the major units to which a communication could be passed as shown in Figure 10.

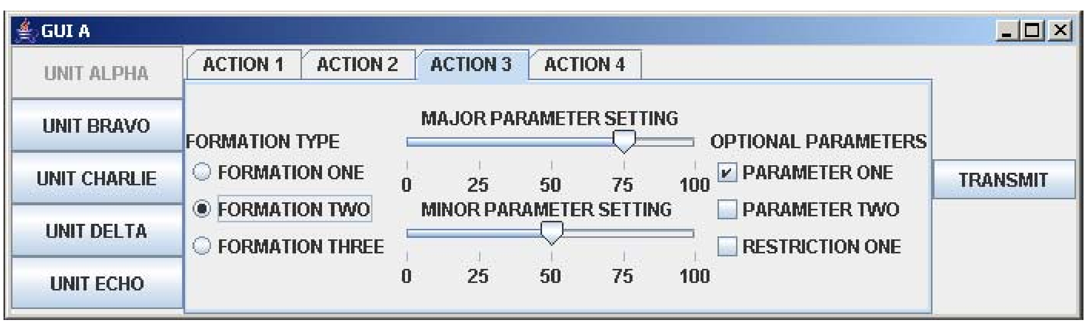


Figure 10. First GUI design for usability study.

Staggers and Kobus dismiss text-based interfaces. They found that nurses' entry of computerized treatment orders improved in task time, correctness, and accuracy while they used an iconic display.[5] A text-based entry for unit commands might be feasible; however, the proclivity for typing errors when working quickly would necessitate an on-screen help menu. A help menu that lists permitted values does the same thing as an iconic display. If a user must frequently reference a help menu to look up a discrete list of values, then those values should be right on the interface anyway.

Rivera and Eng investigated the effect of giving a user too much information at once. Their study on a customizable interface and its effect on perceived mental workload yields criteria for task management displays regarding the number of pieces of information with which a user may effectively cope. They observed that performance declines when a user is forced to sort through irrelevant data placed alongside valid GUI command choices.[6] This appears to be another argument for modal displays. Heeding this advice, we chose to implement the list of unit action commands according to context for both test GUIs.

Wickens, Lee, Liu, and Becker describe the cognitive process by which users apply the concepts of physical location to areas of an interactive display. They suggest that interfaces dealing with discrete groups of representative units should be organized to allow a user to visit each group to the exclusion of others.[7] Kurtenbach, Fitzmaurice, Owen, and Baudel discuss their revolution of the “Hotbox” toolbar used in the modeling tool *Maya*®. In order to embed several hundred commands and make them easily accessible with a minimum number of user actions, they developed a modal interface that grouped functionally similar items into zones.[8] This approach seemed ideal for use in a command and control interface; if the “Hotbox” zones represented units instead of control modes, users could access unit commands by right-clicking on the unit. As displayed in Figure 11, we based the second test GUI on this approach, mimicking the *Maya*® interface.

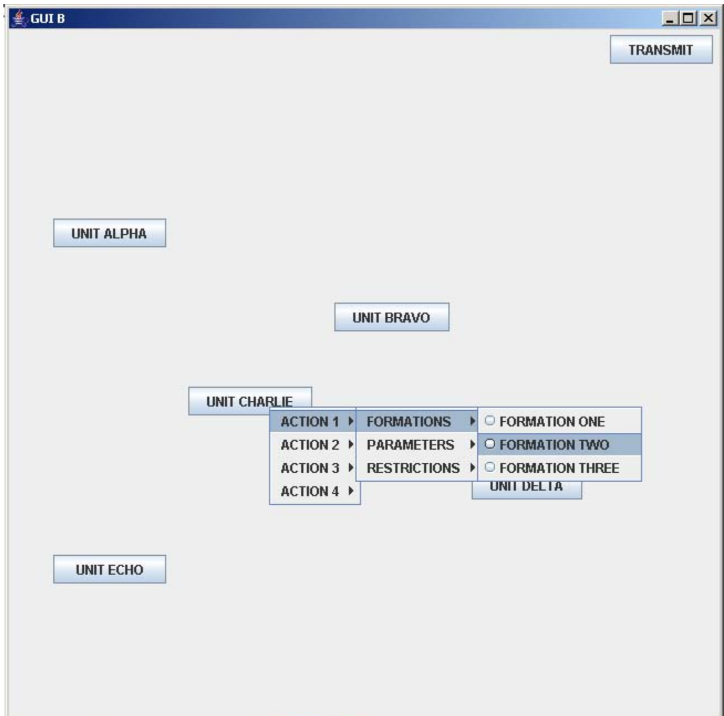


Figure 11. Second GUI design for usability study.

c. Hypothesis

For the usability study, our hypothesis was that we would see no difference in user performance or preference when using either GUI. Both were influenced by current industry practices, and each avoids the characteristics of low-performance GUIs documented in research. It would be a trivial experiment to create a good design and a poor design, and then to test each for performance; our goal was to find if two GUIs built according to best practices were equally adept at facilitating unit command transmission. We chose to measure performance as a ratio of task time multiplied by error rate, and to measure preference by written participant commentary. The alternate hypothesis was that some difference exists between the two interfaces, and that the use of one would result in higher performance than the other.

d. Method

The participants were 24 graduate students (1 woman, 23 men) between the ages of 26 and 42 attending the Naval Postgraduate School. The mean age of participants was 33.5 years. This was a convenience sample, and participants were recruited without compensation. The target demographic for a command and control CBT was a company or field grade military officer possessing a basic familiarity with GUIs; all participants fit this description. Participants were treated in accordance with APA Guidelines for the Ethical Treatment of Human Subjects.

We used a 2(Type of GUI) x 2(which GUI was used first) within-subjects design to explore performance of interface as a function of GUI design. Participants were randomly assigned to one of two groups. We gave pre-test questionnaires to participants; these included questions about familiarity with computer interfaces in general, amount of sleep the participant had during the night before, whether the participant had eaten a meal prior to the experiment, and various demographic data such as age, nationality, and gender. The aim of the pre-test questionnaire was to document as many potential confounding factors as possible. We predicted it would be useful if a need for blocking by any of these factors became necessary.

We needed two computers for testing. Operating system was irrelevant due to the portability of Java, although we chose Microsoft® Windows XP® for convenience. Any processor capable of running a mainstream office productivity application was sufficient to run both GUIs. Participants required a mouse to manipulate the interface, but did not need a keyboard. For our tests, the text displayed on the GUI used Arial capital font of size 12. Each GUI program tracked all button and menu selections; it recorded the total time a participant took to complete a test and output the data to a log file.

We scheduled participants to meet with us in groups of two. At that time, they completed the pre-test questionnaire, and we recorded the time and day of the week. Participants were first given a short tutorial; for each GUI we demonstrated executing the command "Tell Unit Alpha to execute Action 3 with the following parameters: Formation Two, Major Parameter at 75, Minor Parameter at 50, and with Optional Restriction One." We informed the participants that both time and correctness in transmitting orders would be used to calculate performance. After a participant indicated understanding of how the GUI functioned, he or she received a single sheet of paper with an instruction similar to the one given during the tutorial. On initiation, each GUI showed a blank frame with a single button labeled "START." A participant began the test by clicking that button, and ended a test by clicking another button labeled "TRANSMIT." The program recorded time between those two clicks. Participants were not able to see menu choices until they clicked the start button, and choices were hidden again after clicking the transmit button.

Each group of 12 participants completed 10 tasks on each GUI. Group A used GUI A first, and then GUI B. Alternatively, group B used GUI B first, then GUI A. Participants were given a new instruction sheet as soon as they indicated they were ready, with a two minute break between tests for different GUIs.

Following completion of all 20 tests, participants completed a survey; the survey used asked for preference of GUI in terms of intuitiveness, organization, aesthetics, quickness, consistency, enjoyment, frustration, preference, ease of use, and clarity. Upon completion of the questionnaire, participants were thanked for their participation and dismissed.

e. Results

The main findings of the experiment are displayed in Figures 12 and 13, which show mean GUI test time as a factor of which GUI was used. Figure 13 additionally attempts to explain mean test time as an interaction between the factors of which GUI was used first, participant age, and whether the participant took notes during the test. We noted during the study that one user made check marks on the instruction sheets to help track progress; this resulted in much higher times for each of that participant's tests.

The data from log files revealed a significant difference in test times among the two GUIs. For the basic evaluation of GUI used by test time shown in Figure 4, $F(1,46) = 89.085$, $p < .0001$. The more explanatory regression shown in Figure 5, which takes into account additional factors, yields $F(5,42) = 64.970$, $p < .0001$.

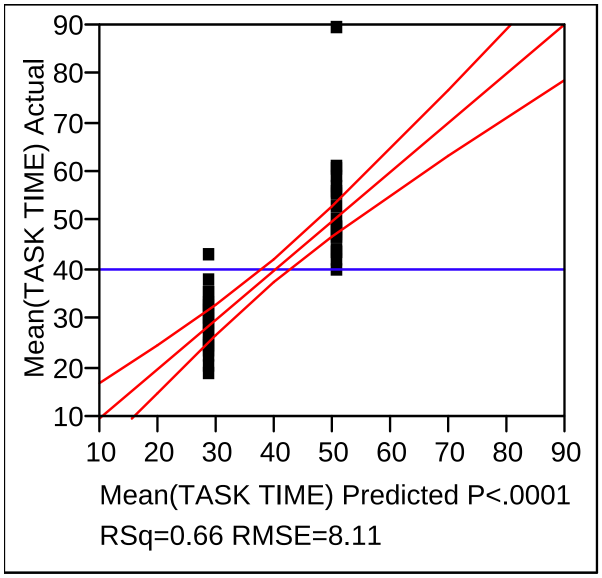


Figure 12. Test time as a factor of GUI used.

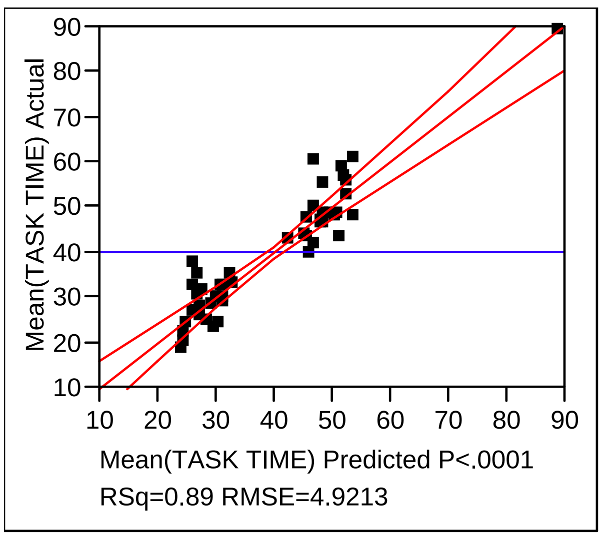


Figure 13. Test time as a result of GUI used, first GUI used, age, and whether the participant took notes to track their test progress.

The post-test questionnaires revealed that of the 24 participants, all 24 preferred the first GUI in terms of intuitiveness, organization, aesthetic quality, quickness of use, consistency, and enjoyment. All but one participant rated the second GUI as the more frustrating of the two, and only two participants chose the second one as the interface with more clarity of design. All 24 participants rated the first GUI as the preferred one.

f. Conclusion

Our goal had been to find the interface that allowed the quickest conveyance of unit instructions. The test results provided an answer. Based on the first test GUI, the communications panel for *Cleared Hot* would consolidate all unit names in one area, while the row list of possible commands would be contextually populated when that unit was selected. Figure 14 shows the result of a minor evolution from the first test GUI to the final version used in the trainer.

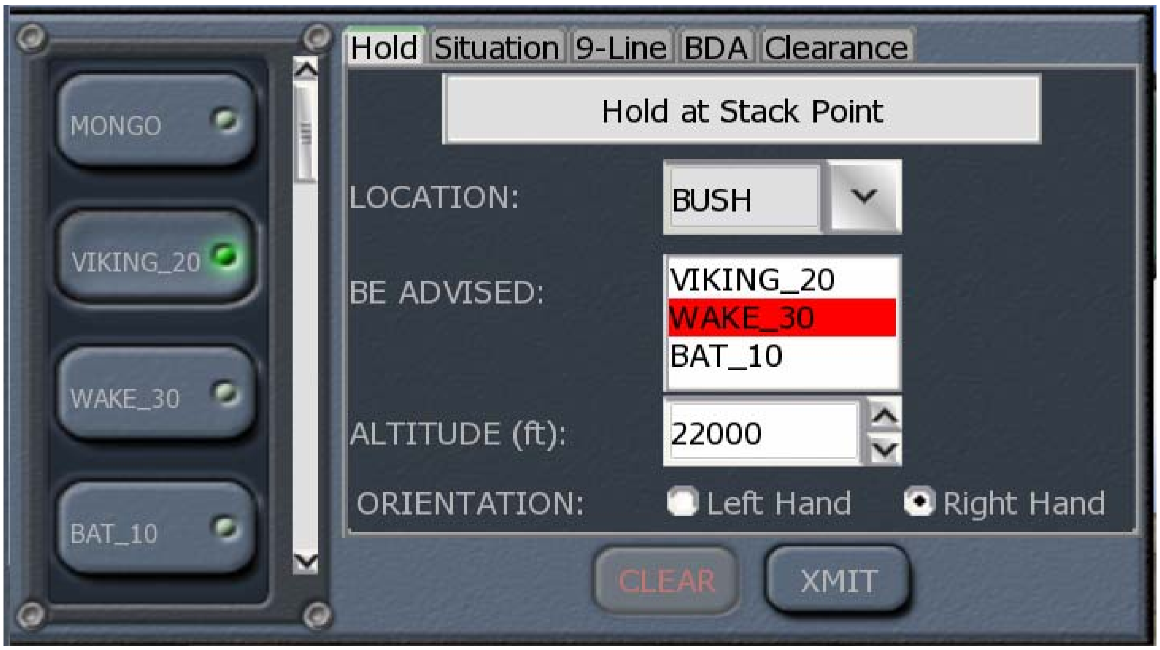


Figure 14. Final Version of Unit Communication Panel in *Cleared Hot*.

Endnotes

1. Marcus, A. 1990. Principles of effective visual communication for GUI design. In *Human-computer interaction: Towards the year 2000*, ed. R. M. Baecker, J. Grudin, W. A. S. Buxton, S. Greenberg, 425-441. San Francisco, CA: Morgan Kaufmann Publishers, Inc.

2. Apple Computer, Incorporated. 1992. *Macintosh human interface guidelines.* Boston: Addison Wesley Professional.

3. M. R. Endsley, B. Bolté, and D. G. Jones. 2003. *Designing for situation awareness: An approach to user-centered design.* New York: Taylor & Francis.

4. J. Whiteside, S. Jones, P. S. Levy, and D. Wixon. 1985. *User performance with command, menu, and iconic interfaces.* Paper presented at the SIGCHI conference on Human factors in computing systems, San Francisco, CA.

5. N. Staggers and D. Kobus. April 2000. Comparing response time, errors, and satisfaction between text-based and graphical user interfaces during nursing order tasks. *Journal of the American Medical Informatics Association*, 7: 164-176.

6. D. Rivera and P. Eng. September 2005. The effect of information overhead on perceived workload and learning. *Proceedings of the Human Factors and Ergonomics Society, USA*, 49: 666-670.

7. D. W. Wickens, J. D. Lee, Y. Liu, and S. E. G. Becker. 2003. *An introduction to human factors engineering* (2nd ed.). Upper Saddle River, NJ: Pearson Prentice Hall.

8. G. Kurtenbach, G. W. Fitzmaurice, R. N. Owen, and T. Baudel. May 1999. *The hotbox: Efficient access to a large number of menu-items.* Paper presented at the SIGCHI conference on Human factors in computing systems, San Francisco, CA.

V. SYSTEM IMPLEMENTATION

A. ARCHITECTURE

As stated in Chapter III, Delta3D is the open source game engine upon which the *Cleared Hot* application is built. In laymen's terms, Delta3D is simply a set of cross-platform libraries which as a standalone entity is not executable. Its design affords end users the flexibility to pick and choose the functionality to incorporate into the larger application. The primary goal of Delta3D is to provide a single, flexible API with the basic elements needed by all visualization applications.[1] In addition to the core components of Delta3D, its modular design facilitates the integration of other well-known open source projects such as Open Scene Graph (OSG), Open Dynamics Engine (ODE), Character Animation Library (CAL), OpenAL, Crazy Eddie's GUI (CEGUI), and others. The complexity of these underlying modules is accessible to users of Delta3D if required; however, should one rather remain oblivious to the behavior of Delta3D external dependencies, successful utilization of the libraries can still easily be achieved.

Cleared Hot version 1.0.4 runs on a Delta3D release that falls somewhere between versions 1.2 and 1.3. Delta3D version 1.2 was released at the end of January 2006, and version 1.3 is expected to be released in the middle of June 2006. The *Cleared Hot* application is written in the C++ language utilizing Microsoft Visual Studio. It relies heavily upon the Delta3D libraries and external dependencies for implementation and visualization of the terrain, game characters, and graphical user interfaces.

Although *Cleared Hot* is a very complex application with over thirty three thousand lines of code, the major functionality may be distilled into three main classes: ClearedHotApp, Game, and RadioMessage. ClearedHotApp is an application class that chooses which game to start dependent upon user input. Its config() function is called by the overall main() function of the application. This config() function subsequently calls the begin() function of an instantiation of a FACAGame, which is a derivative of the interface Game class. Among other things, the FACAGame class is responsible for retrieving the specifics of the user's mission from an XML file and starting the scenario by initializing all the appropriate variables.

The information in the XML file is representative of the typical five paragraph order students receive prior to tactical syllabus events such as FAC(A) flights. Additionally, it contains critical information necessary for initialization of the enemy and friendly entities within the game, such as location, description, and callsign. The intent would be that squadron weapons and tactics instructors generate these XML files for loading into the game in a fashion similar to what is currently done. Ideally, five paragraph orders currently on-hand in a digital format could be parsed by the *Cleared Hot* software and the XML file automatically generated. See Chapter VIII for more detail concerning the proposed *Cleared Hot* Mission Editor.

Once the game variables have been initialized by the FACAGame class, control is returned to the main() function, which subsequently calls the run() function of ClearedHotApp. The application then enters a large loop which contains all the code necessary for updating the user's system based upon their actions.

Communication is a very large and important part of the overall FAC(A) task. Consequently, this thesis focused much attention on the dialogue between entities. RadioMessage is an interface class responsible for ensuring all radio communication messages can be transformed into and out of the string format. The terminology resident in the subclasses of RadioMessage is derived from the personal experiences of several military officers familiar with the FAC(A) task and from numerous USMC doctrinal publications available on terminal attack control.

The CommWidgetController class is also worthy of discussion in the routing of radio messages. It is responsible for assembling the radio messages compiled by the user and sending the messages to the RadioChannel, which is a singleton class utilized to route the messages to all entities listening to the channel. See Chapter VI for a more detailed discussion of the actual radio user interface.

B. ARTIFICIAL INTELLIGENCE (AI)

The artificial intelligence present in *Cleared Hot* is implemented utilizing AI planning techniques, which is a fairly new approach in the gaming community. Typical AI implementation techniques in games solely use finite state machines; however, finite state machines are not very scalable. In an application as complex as *Cleared Hot* where there is not necessarily always a right and wrong sequence of actions to take from one state to another, management of the possible interactions between characters becomes unwieldy. Employment of AI planning tools yields a more tractable problem.

According to Jeff Orkin, one of the developers of the game F.E.A.R., the rapidly growing complexity in games is a problem for all game AI developers and introducing real-time planning was their attempt at solving the problem.[2] Whereas a finite state machine tells an artificially intelligent agent exactly how to accomplish its goals, a planning system tells the agent what its goals and actions are and lets the AI decide how to sequence actions in order to satisfy goals.[3]

C. OVERALL GAME FLOW

Cleared Hot is designed in a manner such that execution of a game sequence closely resembles the events pilots will encounter when supporting typical FAC(A) missions. Successful flights begin with a thorough analysis of the mission; therefore, it is no surprise that preparation of a solid plan for execution is one of the training objectives of *Cleared Hot*. The mission planning phase is replicated subsequent to the user choosing a scenario at the beginning of the game. Selection of the "Ready Room" button on the user interface (UI) presents the user with mission details in the typical five paragraph order format that all Marines are familiar with; however, in order to place more emphasis on the requirement for comprehensive mission planning, the current implementation needs to be more robust. See Chapter VIII for details surrounding proposed feature enhancements to the Ready Room.

Once the user is satisfied with the mission specifics, the game is loaded into the system and the FAC(A) is spawned in the operating area at an instructor selected checkpoint as set forth in the aforementioned XML file containing scenario specifics. The mechanics of transiting from the flight line to the operating area are skipped since those events add minimal training value to the overall FAC(A) task. Once the user is oriented as to his position in the area, he is expected to check in with the Air Officer (AO) in order to receive a situation update and take terminal control. CAS platforms are spawned according to the Air Tasking Order (ATO), also detailed by the instructor in the XML file. It is important to note here that the instructor references are to an "asynchronous instructor in the loop". The instructor is not a requirement for execution of *Cleared Hot*.

The weapons and tactics instructors in the squadron typically maintain several training scenarios for dissemination to FAC(A) syllabus students. Once an instructor loads these scenarios onto a system, the instructor requirement is vacated until time to debrief the student after execution of *Cleared Hot*.

In order to ensure the user travels down the correct path of checking in with the AO and taking terminal control prior to communicating with CAS assets, user actions are limited by the logic implemented in the software. See Appendix B for a graphic depiction of the approved CAS state transitions. The game is put "on rails" because there is training value to providing the user with immediate feedback to actions executed in a sequence that is procedurally inaccurate. The intent is to mitigate the number of potential instances for negative training. Although the WTI would most certainly correct the student on all items executed improperly during the debrief, the decision was made to avoid delay of this critique whenever feasible.

Once the FAC(A) has taken terminal control from the AO, CAS assets immediately check in to receive holding instructions and any applicable updates to the situation as determined by the FAC(A). As the user's situational awareness increases due to his personal reconnaissance of the terminal area, he will mentally start to organize the battle space and formulate attack packages. This training objective is achieved, in part, through implementation of the kneeboard UI, which will be discussed later. The user compiles attack packages on the kneeboard for later dissemination to CAS assets via the radio interface. It is not the construction of attack packages on the kneeboard that sets the CAS AI in motion. When the user is ready, the radio UI is utilized to transmit 9-lines to CAS assets and call-for-fire requests to indirect fire agencies. Some rudimentary validation of the attack packages is conducted before commencement of the run on the target, such as given typical aircraft performance characteristics and adherence to the routing instructions passed by the FAC(A), can the CAS selected for the mission feasibly transit from its present position to the target by the selected time on target (TOT). See Chapter VIII for a more detailed 9-line and call-for-fire validation scheme.

The culmination of a successful attack package is the deliverance of rounds on target and the achievement of the desired effects. In order to achieve that endstate, the

FAC(A) must respond to the CAS assets' "wings level" call with a timely "*Cleared Hot*" call. Type 2 terminal attack control is the sole implementation in *Cleared Hot* version 1.0.4. Although desirable functionality, Type 1 control is not possible because the FAC(A) is unable to visually acquire the attacking aircraft at weapons release. This thesis recommends that both types 1 and 3 be implemented in future versions of *Cleared Hot*. As detailed in Chapter II, the ability to visually acquire the attacking aircraft and discern if the CAS platform is within the prescribed final attack cone before authorizing weapons release is a critical FAC(A) task.

The user is capable of sequencing multiple CAS sorties simultaneously and training with *Cleared Hot* until the decision is made to abort the mission. The implementation of this functionality in version 1.0.4 is one method of achieving the training objective associated with the maintenance of operational tempo. A more intricate plan for allowing the user to quit the game gracefully is discussed in Chapter VIII.

D. GRADING SYSTEM

One may note the absence of a point or grading system in *Cleared Hot*. This is done intentionally for numerous reasons. Evaluation of various aspects of the overall FAC(A) task is very subjective. For example, there is more than one viable option for satisfactorily setting up the battle space geometry for conducting an attack on a target. Although some plans are clearly not tactically sound, such as an attack package void of suppression when an enemy anti-air threat exists in close proximity to the target, ranking the possible combinations of solid plans on a point scale would be arbitrary and meaningless. Additionally, it is not the intent of *Cleared Hot* to be overly judgmental as its designers are not FAC(A) instructors trained by the experts at MAWTS-1. Therefore, since the majority of student pilot learning is achieved in the debrief after one has had the opportunity to reflect on some of the decisions made during the flight, After Action Review (AAR) functionality is critical to facilitating this exchange between student and instructor. It would be prudent for future versions of *Cleared Hot* to implement such a feature in order to increase its chances for positive training transfer.

Nevertheless, there are several objective parameters, or metrics, that could be measured in the game, such as evaluation of certain procedural items, the proper formulation of specific radio calls, and the elapsed time from target identification to engagement. Where resources permitted, these metrics and others were incorporated into the current release of *Cleared Hot*. Additional ideas for immediate measure of a student's performance can be found in Chapter VIII.

E. FEATURES

The focus of the ensuing discussion is on the rationale behind the implementation of the major game functionality. A more detailed description of all *Cleared Hot* features appears in Appendix D.

1. Mini-map

The two dimensional mini-map is designed to simulate the actual paper map that pilots keep in the cockpit for orientation and navigation purposes in flight. Its implementation is necessary in facilitating the user's visualization and understanding of the battle space. The locations of known friendly and enemy assets and preexisting airspace control measures are overlaid on the mini-map for easy reference. Additionally, the FAC(A) navigates the terminal area by laying waypoints on the mini-map. This interaction between the user and the mini-map in the *Cleared Hot* virtual world is directly analogous to the non-flying pilot analyzing the map before dictating a proposed route of flight to the flying pilot in the real world.

2. Kneeboard

The kneeboard is representative of the genuine kneeboard issued to student pilots during primary flight training in Pensacola, Florida. In reality, it is simply a clipboard maintained on the pilot's knee that is capable of holding any number of things, typically a pad of blank paper and notes from the mission brief.

a. 9-line

It is critical that the UI for this feature be designed from a human computer interface (HCI) standpoint because the accurate and timely generation of 9-lines is a focal point in the achievement of commander's intent. Additionally, this thesis predicts that this functionality will be utilized frequently in the execution of missions, thus warranting its careful design. One key design decision that occurred late in the development process was the removal of the 9-line remarks from the radio interface and incorporation of the remarks onto the kneeboard. This allows the user to create the entire 9-line with one UI. It also facilitates the solution to the problem of how to bind the 9-line basic elements and remarks to a particular CAS asset with the user utilizing two different interfaces. The result is one 9-line "object" that is persistent on the kneeboard, available for use in re-attacks when required.

b. Call for Fire (CFF)

The CFF tab on the kneeboard is utilized for developing requests for indirect fire. It is designed in a very similar fashion to the 9-line tab in order to make things intuitive to the user and thus avoid any confusion between the mechanics necessary for creation of the two requests. SEAD is the only operable CFF mission type available to the user in *Cleared Hot*'s current implementation. As resources were limited for this project, when prioritizing features, SEAD was ranked high on the list due to the importance of the combined arms application training objective.

3. Radar View

The radar view is designed to replicate the magnetic course indicators inherent in all military aircraft. Its incorporation into the user's mission console, the lower portion of the *Cleared Hot* UI, is necessary in aiding the user in the critical task of ownship relocation. The ten kilometer distance indicative of the outer range of the radar view is chosen to illustrate to the FAC(A) the surrounding area that should be of immediate concern. Friendly and enemy entities in the world that are outside of that range are clamped to the outside indicating their distance is greater than ten kilometers, yet their specific range is unknown. The predicted impact of this implementation is a student more capable of compartmentalization because he has a better understanding of the battle space than one who is easily saturated by trying to process too much information at once.

The mere existence of friendly and enemy icons on the radar view is cause for debate. Friendly, or "blue force", trackers currently exist in the real world; however, the authors of this thesis are not aware of an analogous system for digital enemy tracking in the Marine Corps. Consequently, making the user omniscient could be viewed as a potential for negative training. The entities' presence on the mini-map is a contributor to the user's omniscience as well; however, the radar view and mini-map are both sterile in the beginning of the game. Game characters are not visible until after the FAC(A) receives the situation update from the AO, representing an increase in user situational awareness. This is another funneling technique designed to push the user down the desired path to a successful training evolution. Additionally, with a tiered approach to training (See Chapter VIII) and some feature enhancements, this functionality could be utilized at lower levels by the novice user and rendered inoperable at higher levels for the more experienced user.

4. Scope

The scope feature is designed to aid the user in examining units on the battlefield without compromising ownship position. This is achieved through a zoom feature resident in most cockpit heads-up displays (HUD). A track feature is implemented allowing the user to lock the scope view onto a target and keep it in sight while navigating from waypoint to waypoint. The AH-1W gimbal limits are imposed on the scope in order to provide realism. Additionally, laser range finding capability is implemented on this UI to provide the user with a ten-digit magnetic grid reference for accurate attack package creation. When in use, the scope clobbers the out-the-window view. This too is realistic in that one cannot look at a map or kneeboard while focusing attention on a HUD.

Endnotes

1. Delta3D. *Features*. (n.d.). Retrieved May 5, 2006, from http://www.delta3d.org/article.php?story=20051209133127695&topic=docs.

2. Orkin, J. (n.d). *3 States and a Plan: The AI of F.E.A.R.* [PowerPoint brief].

3. Ibid.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. VALIDATION PROPOSAL

A. NECESSITY

The development of *Cleared Hot* did not reach a stage where we felt it should be used in an experiment. Although it allows the creation and execution of attack packages, without certain desired features it falls short of the battlefield management trainer we envisioned. Prominent among these is a timeline tool to assist in visualizing operational tempo, 3D rendering of Fire Support Coordination Measures, and options to conduct missions other than SEAD with indirect fire assets.

Detailed in the chapter following this is our visit to MAWTS-1 where we presented the trainer as a proof of concept. There we received acknowledgement of being on the correct path; however, validation does not come from a handshake. Early in development, we intended to conduct a study of the trainer's effectiveness using as participants pilots ready to begin the FAC(A) flight syllabus. We generated a test protocol at that time; this chapter contains that protocol, and it is our hope that it may be used as a guideline for testing when the system reaches further maturity.

B. PARTICIPANTS

The study should consist of volunteers from U.S. military squadrons that conduct the mission of FAC(A). The protocol that follows assumes the participants will be from USMC squadrons, although because Cleared Hot was designed according to joint doctrine, pilots from other services will find nothing unfamiliar in it. Participants should be preparing to begin the FAC(A) flight syllabus; the idea is that they will be familiar already with the applicable texts.

There are four active duty Marine helicopter squadrons on the west coast that conduct FAC(A); at any given time three at most will be at Camp Pendleton, California. We recommend that participants be drawn from these three squadrons, as they are geographically closest to the Naval Postgraduate School; however, two squadrons will also be available on the east coast should the test require a larger subject base.

C. HYPOTHESIS

One method of war-gaming FAC(A) exists in the realm of Ready Room discussions; much education may be gained through a discussion of What if? with senior pilots. The strength of Cleared Hot lies in its ability to present a war-gaming sand box. Using it, a user may attempt various plans of attack to accomplish the mission task; the repercussions of both good and bad plans provide immediate visual feedback. Through such experiments, and with the aid of a structured debrief, a FAC(A) syllabus pilot is afforded the opportunity to learn from his/her successes and failures before they become costly while airborne.

Our hypothesis is that FAC(A) syllabus pilots who receive training on the Cleared Hot trainer, as per the prescribed treatment, will have a greater understanding of FAC(A) procedures, specifically in the areas of battle space management, effective use of fire support assets, and mission communications. Significant positive training will be demonstrated through the use of inventory and exit written and oral debriefs. The null hypothesis is that no significant training will be evidenced.

D. PROTOCOL

1. Background

The experiment has a within-subjects design. Each FAC(A) syllabus pilot will be presented with an inventory exam prior to beginning use of the Cleared Hot trainer. Exam questions will be based upon input from MAG-39 WTIs in order to concentrate on areas of tactical knowledge which they have deemed appropriate for stage. In addition, the inventory exam will query the pilots' familiarity with computers and computer trainers.

For the actual Cleared Hot training sessions, each user is presented with a mission scenario and planning documents similar to those available during actual mission planning. Cleared Hot offers a 3D environment using the Twenty-Nine Palms terrain database, in which both enemy and friendly forces are present as dictated by the mission scenario. The trainer offers no criticism of a participant's actions. Users will be free to take any actions they deem appropriate in support of the mission statement and the Ground Combat Element commander's intent. They will be able to formulate any plan of attack, tactically sound or unsound, and use CAS and indirect fire assets as they see fit. The Cleared Hot trainer will record all actions made during a training session and save the data in a file that may be played back, paused, and rewound in a manner similar to using a VCR. As with debrief and critique of student actions during actual missions, those of the simulated missions will reside with squadron instructors.

2. Support Requirements

a. Optimal

WTI support consists of 2.25 man-hours of time for each FAC(A) syllabus pilot in their squadron, up to a maximum of two FAC(A) syllabus pilots. Ideally, two FAC(A) syllabus pilots are available per squadron, which equates to two WTIs required per squadron, each of which would support the experiment by debriefing one FAC(A) syllabus pilot for 45 minutes (following that pilot's use of Cleared Hot) on each of three days: Tuesday, Wednesday, and Thursday.

FAC(A) syllabus pilot support consists of 7.5 man-hours for each FAC(A) syllabus pilot in a squadron, up to a maximum of two FAC(A) syllabus pilots. Ideally, this would mean two FAC(A) syllabus pilots per squadron, each of which would support the experiment by attending the following sessions:

<u>Monday</u>:
45 minute presentation
45 minute inventory test
<u>Tuesday</u>:
45 minute trainer familiarization use
45 minute test use
45 minute WTI debrief
<u>Wednesday</u>:
45 minute test use
45 minute WTI debrief
<u>Thursday</u>:
45 minute test use
45 minute WTI debrief
<u>Friday</u>:
45 minute follow up test

DURATION	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY
0800 45 minutes	Squadron 1 Presentation	Squadron 1 Pilots use *Cleared Hot* (Familiarization)	Squadron 1 Pilots use *Cleared Hot* (Second use)	Squadron 1 Pilots use *Cleared Hot* (Third use)	Squadron 1 Written Test (Follow Up)
0845 45 minutes	Squadron 1 Written Test (Inventory)	Squadron 1 Pilots use *Cleared Hot* (First Use)	Squadron 1 WTI / pilot debriefs (Second use)	Squadron 1 WTI / pilot debriefs (Third use)	
0930 45 minutes		Squadron 1 WTI / pilot debriefs (First use)			Squadron 2 Written Test (Follow Up)
1100 45 minutes	Squadron 2 Presentation	Squadron 2 Pilots use *Cleared Hot* (Familiarization)	Squadron 2 Pilots use *Cleared Hot* (Second use)	Squadron 2 Pilots use *Cleared Hot* (Third use)	
1145 45 minutes	Squadron 2 Written Test (Inventory)	Squadron 2 Pilots use *Cleared Hot* (First use)	Squadron 2 WTI / pilot debriefs (Second use)	Squadron 2 WTI / pilot debriefs (Third use)	Squadron 3 Written Test (Follow Up)
1230 45 minutes		Squadron 2 WTI / pilot debriefs (First use)			
1400 45 minutes	Squadron 3 Presentation	Squadron 3 Pilots use *Cleared Hot* (Familiarization)	Squadron 3 Pilots use *Cleared Hot* (Second use)	Squadron 3 Pilots use *Cleared Hot* (Third use)	
1445 45 minutes	Squadron 3 Written Test (Inventory)	Squadron 3 Pilots use *Cleared Hot* (First use)	Squadron 3 WTI / pilot debrief (Second use)	Squadron 3 WTI / pilot debrief (Third use)	
1530 45 minutes		Squadron 3 WTI / pilot debrief (First use)			

Table 1. Optimal Support Schedule

b. Minimum

WTI support consists of 1.5 man-hours of WTI time for each FAC(A) syllabus pilot in their squadron, up to a maximum of two FAC(A) syllabus pilots. Ideally, two FAC(A) syllabus pilots are available per squadron, which equates to two WTIs required per squadron, each of which would support the experiment by debriefing one FAC(A) syllabus pilot for 45 minutes (following that pilot's use of Cleared Hot) on each of two days: Tuesday and Wednesday.

FAC(A) syllabus pilot support consists of 6.0 man-hours of FAC(A) syllabus pilot time for each FAC(A) syllabus pilot in a squadron, up to a maximum of two FAC(A) syllabus pilots. Ideally, this would mean two FAC(A) syllabus pilots per squadron, each of which would support the experiment by attending the following sessions:

Monday:
45 minute presentation
45 minute inventory test
Tuesday:
45 minute trainer familiarization use
45 minute test use
45 minute WTI debrief
Wednesday:
45 minute test use
45 minute WTI debrief

Thursday:
45 minute follow up test

DURATION	MONDAY	TUESDAY	WEDNESDAY	THURSDAY
0800 45 minutes	Squadron 1 Presentation	Squadron 1 Pilots use *Cleared Hot* (Familiarization)	Squadron 1 Pilots use *Cleared Hot* (Second use)	Squadron 1 Written Test (Follow Up)
0845 45 minutes	Squadron 1 Written Test (Inventory)	Squadron 1 Pilots use *Cleared Hot* (First Use)	Squadron 1 WTI / pilot debriefs (Second use)	
0930 45 minutes		Squadron 1 WTI / pilot debriefs (First use)		Squadron 2 Written Test (Follow Up)
1100 45 minutes	Squadron 2 Presentation	Squadron 2 Pilots use *Cleared Hot* (Familiarization)	Squadron 2 Pilots use *Cleared Hot* (Second use)	
1145 45 minutes	Squadron 2 Written Test (Inventory)	Squadron 2 Pilots use *Cleared Hot* (First use)	Squadron 2 WTI / pilot debriefs (Second use)	Squadron 3 Written Test (Follow Up)
1230 45 minutes		Squadron 2 WTI / pilot debriefs (First use)		
1400 45 minutes	Squadron 3 Presentation	Squadron 3 Pilots use *Cleared Hot* (Familiarization)	Squadron 3 Pilots use *Cleared Hot* (Second use)	
1445 45 minutes	Squadron 3 Written Test (Inventory)	Squadron 3 Pilots use *Cleared Hot* (First use)	Squadron 3 WTI / pilot debrief (Second use)	
1530 45 minutes		Squadron 3 WTI / pilot debrief (First use)		

Table 2. Minimal Support Schedule

c. Both Schedules

The experiment will last one work week, Monday through Friday. Support from each squadron Operations Officer will be needed to ensure WTIs and FAC(A) syllabus pilots are available and scheduled detailed in the protocol.

A projection system will be needed for either schedule. Under the assumption that each squadron is likely to own a projector, request usage of it for a period of instruction lasting 45 minutes using the squadron's Ready Room or a suitable office space.

d. Testing

FAC(A) syllabus pilots will take two written exams. The first, an inventory test, will be given to the pilots on the first day of the week. After the pilots have worked with Cleared Hot for each of three (two for the minimum support protocol) days during the week, they will be issued a second exam on the last day of the week.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. CONCLUSIONS

A. INTRODUCTION

Original goals of this thesis were to develop an open-source FAC(A) concepts trainer, conduct a fleet evaluation of the product, make the recommended improvements, and subsequently freely distribute the software; however, as the project progressed and time, manpower, and technology constraints took their toll, it became clear that this lofty goal was unrealistic. Consequently, this thesis was bounded, version 1.0.4 was released with basic functionality, and steps were taken to lay the foundation for realization of the original goal. Details regarding proposed feature enhancements for incorporation into a more robust release of *Cleared Hot* are discussed in the next chapter, and the framework for evaluating that trainer was laid out in Chapter VI.

For a proof of concept of the current release of *Cleared Hot* and in lieu of conducting a comprehensive training transfer study, we elected to consult the FAC(A) authorities at MAWTS-1. In May of 2006, the authors of this thesis met with an experienced FAC(A) instructor and both the department head and rotary wing offensive air support specialist from the Aviation Development Tactics and Evaluation (ADT&E) department to discuss the merits of *Cleared Hot*. The current functionality was demonstrated, future enhancements were proposed, and the MAWTS-1 instructors openly provided their unbiased evaluation of our effort.

B. SUCCESS

The overwhelming sentiment from the MAWTS-1 representatives was positive in regards to the trainer's ability to aid in bridging the gap between textbook learning and the in-flight practical application of that knowledge for the pilot just embarking upon the FAC(A) syllabus, our target audience. The novice FAC(A) needs to focus on the fundamentals of the task; therefore, "simple is better" is a good software design methodology. For example, the air tasking order for the mission in version 1.0.4 has no more than two sections of CAS on station at a given time. Forcing the beginner FAC(A) to simultaneously control more assets than that will be counter-productive. The complexity proposed for future versions of *Cleared Hot* needs to be tempered with this thought in mind; the implementation of tiers, or levels, of difficulty is one solution.

Throughout the course of the meeting, our assessment of the manpower limitations in fleet squadrons was reinforced. Consequently, it was no surprise that the ability to operate *Cleared Hot* without an instructor was highlighted as a strength. The inclusion of a robust AAR system in subsequent releases of *Cleared Hot* will greatly complement this stand-alone capability by affording an instructor the opportunity to critique a student's past performances and thus ensure the trainer is not enabling the formulation of bad habit patterns. Additionally, a more extensive implementation for providing immediate feedback to the user is discussed in the next chapter as another means of disallowing poor behavior to go unchecked.

C. RECOMMENDATIONS

The majority of the recommendations provided by the MAWTS-1 instructors focus on desired additions to functionality, implying that the current version of *Cleared Hot* is "not far off" the mark.

1. Mini-map

Pilots should be able to resume routing after selection of the hover button, which stops forward movement. In the current implementation the waypoints are deleted upon clicking of the hover button. Additionally, as tactics, weather, and aircraft configuration sometimes dictate that the FAC(A) orbit instead of hover, this functionality needs to be added as well.

The icons representing air assets, although derived from doctrinal publications, need to be modified in order to be compliant with the typical clip art familiar to the majority of Marine Corps pilots. See Figure 5 for examples.

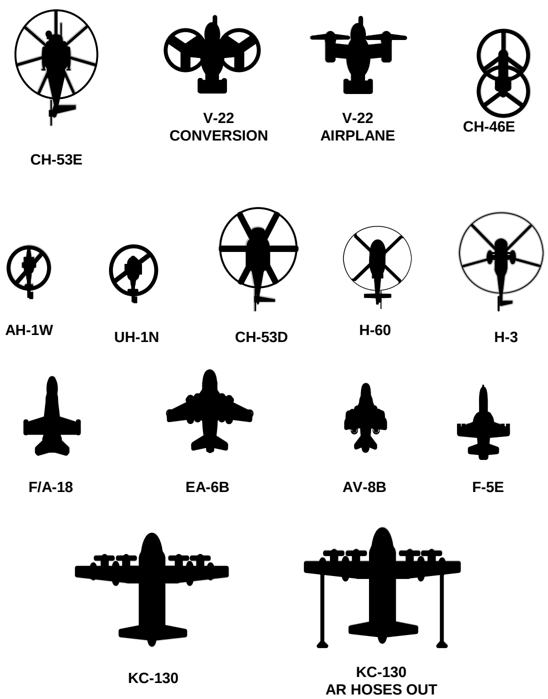


Figure 15. Clip art

Retrieving grid coordinates for targeting information off the mini-map is difficult because the labeling only exists in certain locations on the map. There are workarounds in the real world, such as personally adding grid identifiers closer to the terminal area on one's map. Since some of these workarounds do not easily map to solutions in the virtual world, the user must be given other tools for resolving these problems. In this particular case, tagging the cursor with a grid in the Magnetic Grid Reference System (MGRS) format as it moves around the mini-map is a viable solution.

2. Kneeboard

The operation of the convert button for meter to nautical mile conversions on the 9-line interface is not intuitive. Since pilots typically do the conversions mentally**,** nothing would be lost if this feature was deleted from future releases.

In order to get the student in the habit of ensuring CAS aircraft are properly deconflicted from the flight of artillery, mortar, and naval gunfire rounds, a stay above/below option needs to be implemented in the restrictions portion of the 9-line.

Forward air controllers pass final attack cones to supporting aircraft as two headings; therefore, two text boxes should exist in the 9-line implementation. Currently, only one text box exists and the computer does the math.

3. Radar View

The presence of friendly and enemy force locations on the radar was cause for some debate. As discussed in Chapter V, this feature provides the FAC(A) with an artificially high level of situational awareness. In reality, the FAC(A) is only aware of reported locations of enemy assets unless physically setting eyes on the targets and noting their location. Therefore, the friendly and enemy icons should not be visible on the radar in the more challenging tiers of *Cleared Hot*. A tier two implementation would only show blue and red forces on the mini-map. At tier three, blue and red forces are only visible on the mini-map if the FAC(A) personally scopes, classifies, and identifies them as such. An alternate implementation removes the radar view functionality in its entirety and docks the mini-map in its place on the mission console. The magnetic course indicator symbology would be overlaid on the mini-map in order to facilitate continued support of ownship navigation.

4. Scope

The accuracy of cursor placement required of users in tracking targets of great distances warrants the implementation of functionality more conducive to delicate cursor movements, for example utilization of the arrow keys in conjunction with the shift or ctrl key.

The user should not be limited to tracking hardware on the battlefield; one should be allowed to track any pixel underneath the cursor when selecting the track button.

5. Radio

The UI on the CAS hold tab for advising assets of other players in the area is not intuitive. In the future, an alternate interface should be designed and tested.

To further immerse the user in the mission and facilitate evaluation of his situational awareness at the time, the situation update as passed from the FAC(A) to CAS should not be hard-coded. Utilizing GUI-based entry, the user should be allowed to create the report based off the information currently at his disposal. GUI-based entry is preferred to free text entry in order to provide some realistic bounds on the radio transmission. With a tiered system, the content of the user's situation update could be parsed and used to affect the CAS platforms' situational awareness on higher levels. On lower levels, the transmission is simply reproduced in the communication dialog window, yet the user benefits from the personal creation of the report.

6. Communication

The dialog between entities in the current release of *Cleared Hot* is largely based off the FAC(A) handbook dated 1 January 2004.[1]; however, MAWTS-1 is currently teaching students strict adherence to the JCAS format as set forth in Joint Publication 3-09.3.[2] Some highlights of changes necessary before future releases of *Cleared Hot* are as follows: (1) The FAC(A) must provide the type of control in effect as part of the brief to CAS players prior to commencement of any attacks. (2) CAS aircraft only provide "IP inbound" calls if requested verbally or digitally by the FAC(A). (3) The "wings level" call is replaced with an "in from the (insert cardinal direction here)" or "in (insert magnetic heading here)" call. The joint publication needs to be consulted for compliance with additional mandatory and recommended calls for Types 2 and 3 controls.

7. Other

7.1 All types of CAS control need to be enabled (See Chapter III for justification). Type 1 is currently not possible since the user never visually acquires the attacking aircraft before passing "*Cleared Hot*."

7.2 Laser marking communication sequence between the FAC(A) and CAS aircraft needs to be replicated. See Chapter VIII for more detail.

7.3 Incorporate a pause button for momentarily stopping mission play.

7.4 A map must be made available in the Ready Room with friendly and enemy intelligence derived from the mission specifics.

7.5 Load the terrain database with two additional areas: Yuma ranges and an urban environment.

7.6 Have inclement weather affect game play and force the FAC(A) to better organize the battle space.

7.7 Implement more hot keys for frequently accessed functionality.

7.8 Increase the zoom ratio on scope.

7.9 Look and feel of the scope is presently generic. Allow users to select an airframe at the beginning of the mission. The characteristics of the selected airframe are subsequently rendered in the game.

7.10 Incorporate left and right offset capability on CAS attack runs.

Endnotes

1. Marine Aviation Weapons and Tactics Squadron One. 2004. *Forward air controller (airborne) handbook*.

2. Joint Publication 3-09.3 2005. *Joint tactics, techniques, and procedures for close air support (CAS)*.

THIS PAGE INTENTIONALLY LEFT BLANK

VIII. FUTURE WORK

A. INTRODUCTION

Although resource constraints limited the functionality realized in the implementation of *Cleared Hot* version 1.0.4, several enhanced features were discussed during the design phase of this project. The goal of this chapter is to provide the reader with a roadmap for developing a more robust version of *Cleared Hot*, one satisfying all the critical training tasks determined from the task analysis.

B. AFTER ACTION REVIEW (AAR)

Cleared Hot was never meant to replace the good judgment and experience FAC(A) instructors bring to the table. Therefore, some form of after action review needs to be implemented in order to facilitate an instructor's evaluation of student performance. In *Cleared Hot*'s current form, the instructor must be present looking over the student's shoulder to effectively evaluate and provide guidance. This does not allow the student to maximize idle time because one must now wait until an instructor is free to observe the *Cleared Hot* session. With the inclusion of AAR functionality, the instructor is not needed until debrief time whereby the instructor can then review the student's recorded session and critique the performance.

One option for AAR functionality is Taksi, an open-source video/screen capture tool for 3D applications. According to SourceForge.net, it can capture almost any Windows application using DirectX, OpenGL, or GDI and create an AVI file using any installed VFW codec.[1] Taksi version 0.5 was successfully utilized during our MAWTS-1 proof of concept. Although the authors of this thesis did not experiment with them, newer versions of Taksi have recently been released.

Event logging is another viable option for recording student's actions for subsequent review by a FAC(A) instructor. Additionally, a timeline tab could be added to the kneeboard. Pilots are typically presented a timeline during mission briefs as part of the mission smartpack; therefore, the presence of a virtual timeline on one's kneeboard would not seem out of the ordinary. ATO events and missions planned/executed by the student would all be depicted on this timeline. See Figure 6. The presence of a timeline or event logging together with some form of video capture tool would greatly enhance *Cleared Hot*'s effectiveness as a bridge between textbook and aircraft.

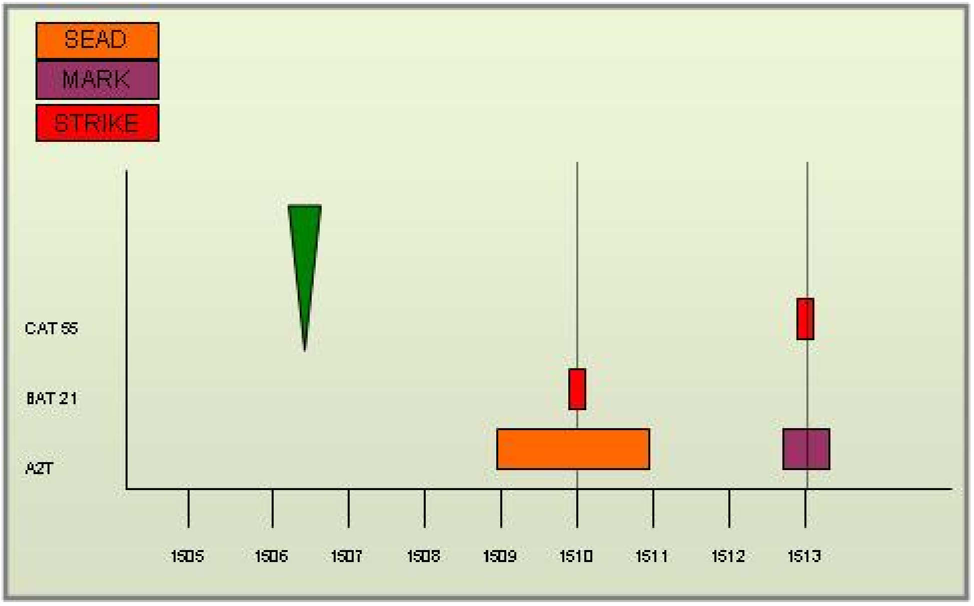


Figure 16. Sample timeline.

C. 3D VISUALIZATION OF THE WORLD

Experience has shown that geometry and management of the airspace are probably the most difficult concepts for young pilots to grasp. Consequently, if one can display to the user, in the virtual environment, how the fire support coordination measures and artillery/mortar round flight trajectories will impact fixed wing aircraft utilization and prosecution of targets, then the hope is that he/she will be better prepared to orchestrate combined arms attacks in the actual aircraft.

For example, should the user transmit a 9-line to CAS, with no restrictions, that conflicts with the artillery gun target line, *Cleared Hot* should warn the user and provide a visual depiction of the intersection of the two trajectories. After evaluating the situation, the user should be afforded the opportunity to either cancel the mission or proceed with execution. Additionally, the user should be able to change viewpoints between the FAC(A) and CAS. Having the ability to observe the battlefield from the CAS aircraft's perspective will enhance the user's overall situational awareness and therefore aid in the decision-making process.

D. TIERS

The introduction of tiers, or levels, into the application will challenge the more experienced *Cleared Hot* users and thus increase the software's value to squadron commanders. It will no longer only be beneficial to the novice FAC(A); refresher pilots can play the game at higher levels and refine their skills without the scaffolding provided at the lower levels. Tier-one functionality has all training aids operable, such as automatic updates to the stack diagram, persistent communication history, FSCM visualizations, changing viewpoints, and accessibility to game player capabilities and limitations. Tier-two functionality could render some or all of those training aids inoperable. Additionally, instructors can introduce measles into the mission at higher tiers, for example bent lasers, no mark, late suppression, and CAS ahead of TOT. The AI running the CAS could also have varying degrees of "smarts" at different tiers.

E. TALK-ON

As discussed previously, the talk-on is a dialogue between the FAC(A) and the CAS pilot whereby the FAC(A) attempts to describe the target area to the CAS pilot by starting the discussion with the largest key terrain feature and subsequently narrowing the scope so the CAS aircrew will be funneled into the target.[2] This is a critical task that needs to be trained; however, open-source voice recognition software and AI capable of solving this problem are currently not available. The problem is that the possible transmissions the FAC(A) could make are infinite; the software would have to respond intelligently to any call.

If the trainer were networked, the FAC(A) could talk via radio directly to the CAS pilot and obtain immediate feedback. Another option involves pausing the game clock during the talk-on and having the user play both FAC(A) and CAS pilot. From the FAC(A) viewpoint, the user would highlight a terrain feature for passing to the CAS pilot. The user would then change to the CAS pilot's viewpoint and attempt to locate and highlight the same terrain feature. If successful, it was a good talk-on terrain feature; if unsuccessful, the user needs to choose another, more prominent feature. The third option involves the development of valid talk-on transmissions, along with some number of distracters, and the insertion of this information into the application possibly through the mission editor. The system would then evaluate the user on the order in which valid radio calls were selected for transmission, such as were terrain features chosen from large to small or small to large. This third option is the least preferred method of implementation as the choices would need to be refreshed after several successful executions, and the burden would then fall on the FAC(A) instructors.

F. STACK DIAGRAM

The stack diagram is a graphical depiction of how the FAC(A) has arranged the CAS assets. The FAC(A) normally handwrites this diagram on his/her kneeboard and updates it manually as necessary. It is a great tool for maintaining situational awareness and visualizing the battle space. The proposed *Cleared Hot* implementation of the stack diagram is described in detail in Appendix D.

G. AIR OFFICER APPROVAL OF THE ATTACK PLAN

According to doctrine, the air officer must approve all FAC(A) generated attack plans before passing to CAS; however, in reality, in order to save time the FAC(A) transmits the attack plan to the CAS pilot over a radio net the air officer is listening to as well. At some point prior to attack plan execution, the air officer relays approval or disapproval over the same radio net. This process is detailed in Figure 7 for incorporation into *Cleared Hot*. The FAC(A) builds the attack plan, transmits the plan to CAS, logic filters are applied to the attack plan, and finally the system alerts the FAC(A) of the air officer's decision. The expected FAC(A) behavior following air officer disapproval is to transmit an abort or cancellation of the attack plan to the CAS. If this does not occur in a timely manner, the air officer aborts the CAS. If this process is implemented correctly, it is not an obstacle to the prosecution of targets and ultimately the user learns that attack plans must be approved before execution.

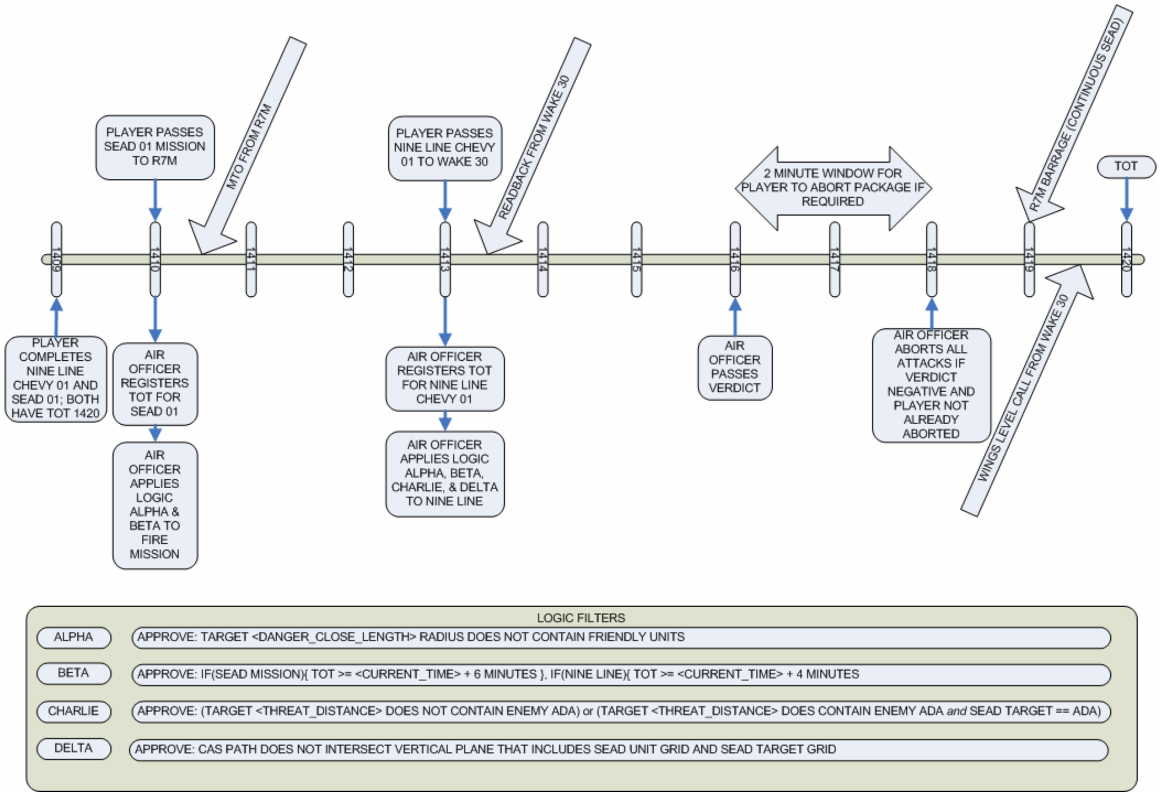


Figure 17. Air Officer Approval of the Attack Plan.

H. LASER DESIGNATION

The use of laser guided bombs is quite prevalent on today's battlefield; therefore, the capability needs to exist in *Cleared Hot*. It is sufficient to simply replicate the communication sequence between the FAC(A) and the CAS pilots in order to achieve this learning objective. Since oftentimes the FAC(A) wingman will laser identify the target, a FAC(A) activated laser button would be inappropriate for this application. Should the CAS aircraft arrive on station with laser guided bombs and the user selects a laser mark on line seven of the 9-line, the appropriate laser communication sequence as set forth in Joint Publication 3-09.3 should be displayed in the communication dialog window.[3]

I. MISSION EDITOR

As stated last chapter, the optimal configuration of *Cleared Hot* contains three terrain databases: 29 Palms, Yuma, and an urban environment. The final release of the application should also contain at least three different mission scenarios: low threat, medium threat, and high threat. Although this proposed configuration will provide a good foundation for squadron FAC(A) instructors, there are an infinite amount of mission scenarios that could be developed for execution, not to mention the ones already in existence in most squadron operations departments. Therefore, the purpose of the mission editor becomes clear; it is to facilitate FAC(A) instructors with incorporating new and challenging mission scenarios to the application. The mission editor should be GUI-based with the capability to parse scenarios already in digital five paragraph order format.

J. MISSION PLANNING MAP

All good plans start with a thorough map analysis of the situation. An editable map should be available to the user during the mission planning phase while viewing the five paragraph order. The map should initially contain the "big picture" with current known friendly/enemy locations, airspace control measures, and arrows showing expected enemy movements. Additionally, the user should be afforded the opportunity to annotate the map with proposed target area flows, battle positions, holding areas, and stack points just as the prudent pilot would do in the real world.

K. FAC(A) PLATFORM CHANGES

Currently, the FAC(A) ownship closely resembles the AH-1W. Since this is not the only platform capable of conducting FAC(A), the option should be given to the user at the beginning of the mission to select ownship, for example AH-1W, UH-1N, and FA-18D. The look/feel and performance characteristics of the selected platform should then automatically be loaded into the game. It is important that the novice FAC(A) be comfortable with his/her environment so focus is not diverted from learning the new task at hand. Along the same lines, *Cleared Hot* should not be solely capable of running fixed-wing CAS. Rotary-wing CAS needs to be incorporated as well.

L. INTEROPERABILITY

The Marine Corps has embraced the advantages training in a virtual world can provide to the warfighter. The Deployable Virtual Training Environment (DVTE)

> ...is a first person skills sustainment trainer that trains Marines by using a simulation network with reconfigurable workstations capable of emulating a variety of weapon systems. Individuals select the weapon, vehicle, or leadership billet desired, then join a virtual battle space where others and synthetic forces are engaged in virtual operations. Individual MAGTF skills can be trained in this virtual environment using a Semi-Autonomous Force (JSAF) model as its basis. The project responds to the need for a

> flexible, DEPLOYABLE, training system that provides combined arms MAGTF and Naval Integration training. Currently a prototype desktop training network, the DVTE prototype addresses a significant subset of USMC combined arms. DVTE provides a custom-built Combined Arms Network (CAN) covering most USMC ground and air weapon systems, and is a USMC critical capability for JNTC participation.[4]

To increase the utility of *Cleared Hot*, it should be augmented appropriately in order to ensure it is interoperable with DVTE and the CAN.

M. AUDIO

In order to fully immerse the user into the trainer, audio needs to be added to the application. Among other reasons, the young pilot will find compartmentalization and multi-tasking easier to achieve if radio calls are audible. In its current implementation, with no audio, the user must focus attention on the communication dialog window in order to read all traffic, lest something critical be missed. Although all radio calls are important, some transmissions are merely informative and require no response from the user. Consequently, *Cleared Hot* should reflect acknowledgement of this fact by making the communication process less intrusive.

N. ADJUST FIRE/FIRE FOR EFFECT

The Call for Fire (CFF) functionality needs to be completed. Although the GUI is complete, the adjust fire and fire for effect features are not operable in *Cleared Hot* version 1.0.4. Figure 8 depicts the proposed flow of events for completion of the CFF task in *Cleared Hot*.

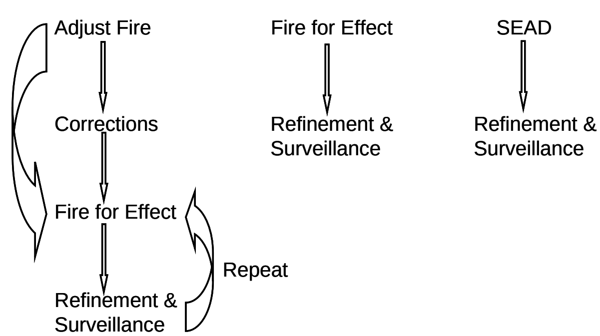


Figure 18. CFF flow of events.

O. BATTLE DAMAGE ASSESSMENT (BDA)

It is critical for the FAC(A) to accurately note and transmit the battle damage assessment to the appropriate parties in order to increase the situational awareness of all involved in the mission. Intelligence personnel also need to know the current enemy disposition so they can better aid commanders in making crucial decisions. A BDA tab is currently implemented on the *Cleared Hot* kneeboard interface; however, it is blank and needs to be properly developed.

Typically, pilots annotate BDA on their kneeboards for subsequent radio relay to CAS, indirect fire units, and other agencies. Therefore, a GUI can be implemented on the BDA tab whereby users can manually input the appropriate information at the more challenging tiers/levels of *Cleared Hot*. Otherwise, the process can be automated at the more basic game levels. The BDA report should include at a minimum: size, activity, location, time, and observed damage.

P. BATTLE HANDOVER (BHO)

Just before the current FAC(A) returns to base after supporting the GCE, a battle handover brief is conducted with either the air officer or the oncoming FAC(A) in order to increase the new terminal controller's situational awareness. Information contained in the brief should generally follow the SMEAC format.

In order to ensure the user learns what is appropriate to report to the new terminal controller, the brief should be manually compiled and transmitted rather than automatically composed by the application. A tab on the *Cleared Hot* kneeboard has already been reserved for BHO composition.

Endnotes

1. SourceForge.net. *Taksi*. 2006. Retrieved July 6, 2006, from http://sourceforge.net/projects/taksi/.

1. Marine Aviation Weapons and Tactics Squadron One. 2004. *Forward air controller (airborne) handbook*.

3. Joint Publication 3-09.3 2005. *Joint tactics, techniques, and procedures for close air support (CAS)*.

4. MARCORSYSCOM. *Program Manager for Training Systems (PM TRASYS)*. (n.d.). Retrieved July 10, 2006, from http://www.marcorsyscom.usmc.mil/trasys/trasysweb.nsf/DocList2/Deployable%20Virtual%20Training%20Environment%20(DVTE)?OpenDocument.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX A. COGNITIVE TASK ANALYSIS

Task ID	Task Description	Critical Task
1	**GOAL:** Conduct FAC(A) mission	
1.1	**GOAL:** Plan mission	
1.1.1	**GOAL:** Gather planning data	
1.1.1.1	METHOD**:** Consolidate operational documents	
1.1.1.1.1	METHOD: Obtain operational documents from MAG FAC(A) Mission Commander or Air Officer	
1.1.1.1.1.1	OPERATOR(M): Obtain Ground Scheme of Maneuver	
1.1.1.1.1.2	OPERATOR(M): Obtain Ground Commander's Intent	
1.1.1.1.1.3	OPERATOR(M): Obtain Specified and Implied FAC(A) Tasking	
1.1.1.1.1.4	OPERATOR(M): Obtain Air Fire Plan	
1.1.1.1.1.5	OPERATOR(M): Obtain Target Area Coordination Plan	
1.1.1.1.1.6	OPERATOR(M): Obtain Fire Support Coordination Measures	
1.1.1.1.1.7	OPERATOR(M): Obtain Expected Area of Operation	
1.1.1.1.1.8	OPERATOR(M): Obtain Supported Unit's Expected Locations	
1.1.1.1.1.9	OPERATOR(M): Obtain Initial Positions of TACPs	
1.1.1.1.1.10	OPERATOR(M): Obtain Fire Support Plan	
1.1.1.1.1.11	OPERATOR(M): Obtain Target Precedence List	
1.1.1.1.1.12	OPERATOR(M): Obtain Reactive Attack Guidance Matrix	
1.1.1.1.1.13	OPERATOR(M): Obtain Fire Support Asset Information	
1.1.1.1.1.14	OPERATOR(M): Obtain SEAD SOP	
1.1.1.1.1.15	OPERATOR(M): Obtain LASER Employment Plan	
1.1.1.1.1.16	OPERATOR(M): Obtain FAC(A) Employment Plan	
1.1.1.1.1.17	OPERATOR(M): Obtain Available CAS Asset Information	
1.1.1.1.1.18	OPERATOR(M): Obtain CAS Asset Control Plan	
1.1.1.1.1.19	OPERATOR(M): Obtain Available FAC(A) Asset Information	
1.1.1.1.1.20	OPERATOR(M): Obtain Available Tanker Asset Information	
1.1.1.1.1.21	OPERATOR(M): Obtain FARP Location Information	
1.1.1.1.1.22	OPERATOR(M): Obtain Routing Information	

1.1.1.1.1.23	OPERATOR(M): Obtain List of Control Points and Initial Points	
1.1.1.1.1.24	OPERATOR(M): Obtain List of Battle Positions and Holding Areas	
1.1.1.1.1.25	OPERATOR(M): Obtain Communication Plan and List of Communication Nets	
1.1.1.1.1.26	OPERATOR(M): Obtain List of Code Words and Pro-Words	
1.1.1.1.1.27	OPERATOR(M): Obtain CASEVAC Plan	
1.1.1.1.1.28	OPERATOR(M): Obtain TRAP and SERE Plan	
1.1.1.1.1.29	OPERATOR(C): Verify Reference Map Datum and Coordinate Format	
1.1.1.1.2	METHOD**:** Consolidate Intelligence from Intelligence Officer or Air Officer	
1.1.1.1.2.1	OPERATOR(M): Obtain Ground Order of Battle	
1.1.1.1.2.2	OPERATOR(M): Obtain Air Order of Battle	
1.1.1.1.2.3	OPERATOR(M): Obtain Missile Order of Battle	
1.1.1.1.2.4	OPERATOR(M): Obtain Enemy Forces Most Likely Course of Action	
1.1.1.1.2.5	OPERATOR(M): Obtain enemy Forces Most Dangerous Course of Action	
1.1.1.1.2.6	OPERATOR(M): Obtain Friendly Forces Situation	
1.1.1.1.2.7	OPERATOR(M): Obtain Maneuver Control Measure Information	
1.1.1.1.2.8	OPERATOR(M): Obtain Main Effort Information	
1.1.1.1.2.9	OPERATOR(M):: Obtain Reconnaissance Unit Information	
1.1.1.1.2.10	OPERATOR(M): Obtain SOF Team Locations	
1.1.1.1.2.11	OPERATOR(M): Obtain ROE Restrictions	
1.1.1.1.3	METHOD: Consolidate Higher Headquarters Operational SOPs from Operations Officer	
1.1.1.1.3.1	OPERATOR(M): Obtain Operations Order	
1.1.1.1.3.2	OPERATOR(M): Obtain Theater and Operations SOPs	
1.1.1.1.3.3	OPERATOR(M): Obtain Air Tasking Order	
1.1.1.1.3.4	OPERATOR(M): Obtain Automated Communication Electronic Operating Instructions	
1.1.1.1.4	METHOD: Consolidate Environmental Data	
1.1.1.1.4.1	OPERATOR(M): Obtain Meteorological Data and Weather Forecast	
1.1.1.1.4.2	OPERATOR(M): Obtain Solar and Lunar Data	
1.1.1.1.4.3	OPERATOR(M): Obtain Electro-Optical Tactical Decision Aid Document	
1.1.1.1.5	METHOD: Determine Aircraft Capabilities	
1.1.1.1.5.1	OPERATOR(C): Review Available Ordnance and ASE with Ordnance Officer	
1.1.1.1.5.2	OPERATOR(C) :Review Available COMSEC Gear with Avionics Officer	
1.1.1.1.5.3	OPERATOR(C): Review Aircraft Discrepancy Books for Mission Detractors	
1.1.1.1.5.4	OPERATOR(C): Calculate Weight and Loading Data	
1.1.2	**GOAL:** Conduct Objective Area analysis	
1.1.2.1	METHOD: Formulate plan for ROE	
1.1.2.1.1	OPERATOR(C): Decide what is the ROE	
1.1.2.1.2	OPERATOR(C): Decide what are the commit criteria	

1.1.2.1.3	OPERATOR(C): Decide when can we pull the trigger	
1.1.2.1.4	OPERATOR(C): Decide what are the weapon conditions in the objective area	
1.1.2.1.5	OPERATOR(C): Decide what weapon conditions conform to the ROE	
1.1.2.1.6	OPERATOR(C): Determine the appropriate authentication / encryption	
1.1.2.1.7	OPERATOR(C): Decide what level of compromise makes us abort	
1.1.2.1.8	OPERATOR(C): Decide what criteria determines the EMCON condition	
1.1.2.1.9	OPERATOR(C): Decide who makes the EMCON condition call	
1.1.2.1.10	OPERATOR(C): Decide what threat constitutes covered communications	
1.1.2.1.11	OPERATOR(C): Decide how to adapt if everyone cannot go covered	
1.1.2.1.12	OPERATOR(C): Decide what AKAC sheet are we using	
1.1.2.2	METHOD: Formulate plan for timing	
1.1.2.2.1	OPERATOR(C): Decide what type of request is being fulfilled	
1.1.2.2.2	OPERATOR(C): Determine if PPOC	
1.1.2.2.3	OPERATOR(C): Decide if TOT or GPS time, how all players will get the hack	
1.1.2.2.4	OPERATOR(C): Decide what is the clearance to fire	
1.1.2.2.5	OPERATOR(C): Determine who shoots first	
1.1.2.2.6	OPERATOR(C): Decide when engagements can begin	
1.1.2.2.7	OPERATOR(C): Decide what is the backup plan	
1.1.2.2.8	OPERATOR(C): Determine what are the signals	
1.1.2.3	METHOD: Formulate plan for dealing with forecasted meteorological conditions	
1.1.2.3.1	OPERATOR(C): Decide if time of day helps or hinders the mission, and why	
1.1.2.3.2	OPERATOR(C): Determine if all aircrew are proficient enough for NVD missions	
1.1.2.3.3	OPERATOR(C): Decide if the wind will mask signatures in the BP	
1.1.2.3.4	OPERATOR(C): Decide which way the illumination / mark will drift	
1.1.2.3.5	OPERATOR(C): Decide if blowing smoke will obscure targets	
1.1.2.4	METHOD: Decide how temperature will affect performance	
1.1.2.4.1	OPERATOR(C): Determine the effect on aircraft performance	
1.1.2.4.2	OPERATOR(C): Determine the effect on weapon performance	
1.1.2.4.3	OPERATOR(C): Decide how LASER will be propagated	
1.1.2.4.4	OPERATOR(C): Decide how will clouds affect goggle performance	
1.1.2.4.5	OPERATOR(C): Decide if clouds will force the FW into the threat	

1.1.2.4.6	OPERATOR(C): Decide how visibility will affect acquisition of targets	
1.1.2.5	METHOD: Decide how visibility will affect the mission	
1.1.2.5.1	OPERATOR(C): Decide if visibility suggests slowing ingress airspeed	
1.1.2.5.2	OPERATOR(C): Decide if visibility suggests moving BPs closer to the threat	
1.1.2.5.3	OPERATOR(C): Decide if visibility suggests making formations tighter	
1.1.2.5.4	OPERATOR(C): Decide if EOTDA suggests moving BPs closer to the threat	
1.1.2.5.5	OPERATOR(C): Decide if humidity will affect FLIR performance	
1.1.2.5.6	OPERATOR(C): Decide what are the sun / moon azimuth, altitude, luminance levels	
1.1.2.5.7	OPERATOR(C): Decide if the sun / moon can used to advantage during ingress / egress	
1.1.2.5.8	OPERATOR(C): Create a plan for using or avoiding shadows	
1.1.2.6	METHOD: Formulate plan for integration / deconfliction	
1.1.2.6.1	METHOD: Review potential route conflicts	
1.1.2.6.1.1	OPERATOR(C): Decide if deconfliction is provided for multiple waves of assaults	
1.1.2.6.1.2	OPERATOR(C): Decide if deconfliction is provided for RW CAS	
1.1.2.6.1.3	OPERATOR(C): Decide if assault gunner fields of fire are deconflicted with CAS & GCE	
1.1.2.6.1.4	OPERATOR(C): Decide if deconfliction is provided for FW & RW CAS fire	
1.1.2.6.1.5	OPERATOR(C): Decide if deconfliction is provided for mutually supporting arms	
1.1.2.7	METHOD: Determine if CAS assets know the scheme of maneuver	
1.1.2.7.1	OPERATOR(C): Decide if deconfliction is provided for the GCE	
1.1.2.7.2	OPERATOR(C): Decide if deconfliction is provided for the assaults	
1.1.2.8	METHOD: Determine the plan for CAP / AAW / OAAW / Reconnaissance	
1.1.2.8.1	OPERATOR(C): Review IFF procedures	
1.1.2.8.2	OPERATOR(C): Review call signs & frequencies in case bandits appear	
1.1.2.8.3	OPERATOR(C): Review location & response time	
1.1.2.8.4	OPERATOR(C): Determine CAP / AAW / OAWW will know who you are	
1.1.2.9	METHOD: Formulate plan for supporting arms:	
1.1.2.10	METHOD: Ensure the GCE knows your plan	
1.1.2.10.1	OPERATOR(C): Review with the GCE your general plan	
1.1.2.10.2	OPERATOR(C): Review with the GCE your fire support plan	
1.1.2.11	METHOD: Review the GCE data	
1.1.2.11.1	OPERATOR(C): Review call signs & frequencies	
1.1.2.11.2	OPERATOR(C): Review range fans	

1.1.2.11.3	OPERATOR(C): Review min / max ordinances	
1.1.2.11.4	OPERATOR(C): Determine if the GCE can range the objective area	
1.1.2.12	METHOD: Review artillery data	
1.1.2.12.1	OPERATOR(C): Determine how many tubes there are	
1.1.2.12.2	OPERATOR(C): Determine types and number of rounds available	
1.1.2.12.3	OPERATOR(C): Determine if they will be there while you're on station	
1.1.2.12.4	OPERATOR(C): Determine if have you spun routing & control measures with the DASC	
1.1.2.13	METHOD: Review contingency procedures	
1.1.2.13.1	OPERATOR(C): Review IFF	
1.1.2.13.2	OPERATOR(C): Review RTF	
1.1.2.13.3	OPERATOR(C): Review Lame Duck	
1.1.2.13.4	OPERATOR(C): Review rendezvous procedures	
1.1.2.13.5	OPERATOR(C): Determine if all aircrew know the GCE scheme of maneuver	
1.1.2.13.6	OPERATOR(C): Determine if FSP is coordinated with follow-on waves & all CAS players	
1.1.2.14	METHOD: Formulate plan for fire support coordination	
1.1.2.14.1	OPERATOR(C): Decide Target lists - what are all designated targets for:	
1.1.2.14.2	OPERATOR(C): Decide Arty	
1.1.2.14.3	OPERATOR(C): Decide naval surface fire	
1.1.2.14.4	OPERATOR(C): Decide Air targets	
1.1.2.14.5	OPERATOR(C): Decide which ones will support your mission	
1.1.2.14.6	OPERATOR(C): Decide how will you contact them (call signs & frequencies)	
1.1.2.14.7	OPERATOR(C): Determine if indirect fire assets have been contacted for coordination	
1.1.2.14.8	OPERATOR(C): Understand Fire support coordination measures	X
1.1.2.14.9	OPERATOR(C): Decide what are all the FSCMs within the objective area	
1.1.2.14.10	OPERATOR(C): Decide how FSCMs affect your plan	
1.1.2.14.11	OPERATOR(C): Determine of the GCE, assaults, FW, etc. know your FSCMs	
1.1.2.14.12	OPERATOR(C): Decide who controls release of ordnance & over what net	
1.1.2.14.13	OPERATOR(C): Decide the criteria for a "cleared hot"	
1.1.2.14.14	OPERATOR(C): Decide what constitutes reasonable assurance	
1.1.2.14.15	OPERATOR(C): Decide what is the no communications fire support plan	
1.1.2.14.16	OPERATOR(C): Decide how target marking will be made	
1.1.2.14.17	OPERATOR(C): Determine location of friendlies	
1.1.2.14.18	OPERATOR(C): Determine location of FLOT	
1.1.2.14.19	OPERATOR(C): Determine location of enemy assets	
1.1.2.14.20	OPERATOR(C): Decide how friendlies will signal CAS aircraft	
1.1.2.15	METHOD: Formulate plan for Target priority	

1.1.2.15.1	METHOD: Decide what are the GCE desires	
1.1.2.15.1.1	OPERATOR(C): Decide Primary	
1.1.2.15.1.2	OPERATOR(C): Decide Alternate	
1.1.2.15.2	METHOD: Decide what are the ACE desires	
1.1.2.15.2.1	OPERATOR(C): Decide Primary	
1.1.2.15.2.2	OPERATOR(C): Decide Alternate	
1.1.2.15.3	OPERATOR(C): Decide which specified targets are a direct threat to the mission	
1.1.2.15.4	OPERATOR(C): Decide whether the GCE & ACE targeting desires match up	
1.1.2.15.5	OPERATOR(C): Decide what action is required when you see targets of opportunity	
1.1.2.16	METHOD: Formulate plan for Flight member responsibilities	
1.1.2.16.1	METHOD: Decide who will do FAC(A)	
1.1.2.16.1.1	OPERATOR(C): Decide Primary	
1.1.2.16.1.2	OPERATOR(C): Decide Alternate	
1.1.2.16.1.3	OPERATOR(C): Decide who navigates:	
1.1.2.16.1.4	OPERATOR(C): Decide enroute to ha:	
1.1.2.16.1.5	OPERATOR(C): Decide Ha to bps:	
1.1.2.16.2	METHOD: Decide objective area egress and RTB:	
1.1.2.16.2.1	OPERATOR(C): Decide Primary	
1.1.2.16.2.2	OPERATOR(C): Decide who passes:	
1.1.2.16.2.3	OPERATOR(C): Decide BDA	
1.1.2.16.2.4	OPERATOR(C): Decide MISREPs	
1.1.2.16.2.5	OPERATOR(C): Decide IFREPs	
1.1.2.16.2.6	OPERATOR(C): Decide To what agency on what freq	
1.1.2.16.3	METHOD: Decide what are all the call signs & frequencies you'll use for:	
1.1.2.16.3.2	OPERATOR(C): Decide Supported agencies	
1.1.2.16.3.2	OPERATOR(C): Decide Supporting agencies	
1.1.2.16.4	OPERATOR(C): Decide if your communications be covered or clear	
1.1.2.16.5	OPERATOR(C): Decide if you use plain language or execution checklists	
1.1.2.16.6	METHOD: Decide what are your inter-flight nets:	
1.1.2.16.6.1	OPERATOR(C): Decide Primary	
1.1.2.16.6.2	OPERATOR(C): Decide Kick 1	
1.1.2.16.6.3	OPERATOR(C): Decide Kick 2	
1.1.2.16.7	METHOD: Decide what are your external nets	
1.1.2.16.7.1	OPERATOR(C): Decide Primary	
1.1.2.16.7.2	OPERATOR(C): Decide Alternate	
1.1.2.16.8	OPERATOR(C): Decide what is your no communication plan	
1.1.2.16.9	OPERATOR(C): Decide who will cover, who will attack, & when will this change	
1.1.2.16.10	OPERATOR(C): Decide what will you do for a downed aircraft	
1.1.2.16.11	OPERATOR(C): Decide who will be the on-scene commander	
1.1.2.16.12	OPERATOR(C): Decide what is TRAP response time, call sign, & freq	
1.1.2.16.13	OPERATOR(C): Decide how downed aircraft affect go	

	– no go	
1.1.2.16.14	OPERATOR(C): Decide whether you attempt hasty trap, and what is criteria	
1.1.2.16.15	OPERATOR(C): Decide whether to continue or delay for downed aircraft	
1.1.2.17	METHOD: Formulate plan for go / no go criteria	
1.1.2.17.1	OPERATOR(C): Decide what type & how many aircraft are required	
1.1.2.17.2	OPERATOR(C): Decide how many troops are needed in zone on the first wave	
1.1.2.17.3	OPERATOR(C): Decide how roles & responsibilities shift as aircraft drop out	
1.1.2.17.4	METHOD: Decide what is required to enter the objective area:	
1.1.2.17.4.1	OPERATOR(C): Determine required fuel	
1.1.2.17.4.2	OPERATOR(C): Determine required ordnance	
1.1.2.17.5	OPERATOR(C): Determine what is required upon leaving the objective area	
1.1.2.17.6	OPERATOR(C): Decide if bingos are based on a FARP	
1.1.2.17.7	OPERATOR(C): Decide what ASE is required to enter the objective area	
1.1.2.17.8	OPERATOR(C): Decide what happens if either ASE or weapon systems fail enroute	
1.1.2.17.9	OPERATOR(C): Decide Must you have crypto / SINCGARS / HAVEQUICK to do the mission	
1.1.2.17.10	OPERATOR(C): Decide what is your single radio & lost communication plan in objective area	
1.1.2.17.11	OPERATOR(C): Decide what threat makes a no go in objective area, & who decides that	
1.1.2.17.12	OPERATOR(C): Decide what determines a hot or cold LZ	
1.1.2.18	METHOD: Formulate a plan for Weaponeering	
1.1.2.18.1	METHOD: Evaluate expected targets in the objective area:	
1.1.2.18.1.1	OPERATOR(C): Determine how many	
1.1.2.18.1.2	OPERATOR(C): Determine what type	
1.1.2.18.1.3	OPERATOR(C): Determine what are their strengths	
1.1.2.18.1.4	OPERATOR(C): Determine what are their critical vulnerabilities	
1.1.2.18.1.5	OPERATOR(C): Determine from your bps, what are their expected aspects	
1.1.2.18.1.6	OPERATOR(C): Determine whether targets are dug in, hardened, & do they have reactive armor	
1.1.2.18.1.7	OPERATOR(C): Decide what effect is needed on each target for mission success	
1.1.2.18.1.8	OPERATOR(C): Decide what type & quantity ordnance are needed for each target	
1.1.2.18.10	OPERATOR(C): Decide what type & quantity ordnance is available	
1.1.2.18.2	OPERATOR(C): Decide what the JMEMs say is needed to get desired results	
1.1.2.18.3	OPERATOR(C): Decide F-pole analysis	
1.1.2.18.4	OPERATOR(C): Determine against an air threat, who wins with our ordnance load	

1.1.2.18.5	OPERATOR(C): Determine against a ground threat, who wins with our ordnance load	
1.1.2.18.6	OPERATOR(C): Decide if terminal area tactics are affected against either air or ground adversaries	
1.1.2.19	METHOD: Formulate plan for FARP	
1.1.2.19.1	OPERATOR(C): Decide if a FARP is available	
1.1.2.19.2	OPERATOR(C): Determine if its location helps us	
1.1.2.19.3	OPERATOR(C): Decide what are the call signs, frequencies, and course rules into the FARP	
1.1.2.19.4	OPERATOR(C): Determine how much fuel & ordnance is available there	
1.1.2.19.5	OPERATOR(C): Determine how many spots exist, & what is the turn-around time	
1.1.2.20	METHOD: Formulate plan for radar threat considerations	
1.1.2.20.1	OPERATOR(C): Determine radar terrain masking profiles for the known threats	
1.1.2.20.1.1	OPERATOR(C): Determine enroute profiles	
1.1.2.20.1.2	OPERATOR(C): Determine objective area profiles	
1.1.2.20.2	OPERATOR(C): Decide how will they affect your safe altitudes	
1.1.2.20.3	OPERATOR(C): Decide when & where we can expect radar warning indications	
1.1.2.20.4	OPERATOR(C): Decide what is the radar resolution cell	
1.1.2.20.5	OPERATOR(C): Decide how will you use the radar resolution cell to your advantage	
1.1.2.21	METHOD: Formulate plan for Holding area / rally points	
1.1.2.21.1	OPERATOR(C): Decide whether you need a holding area	
1.1.2.21.2	OPERATOR(C): Decide if the holding area big enough for all mission aircraft	
1.1.2.21.3	OPERATOR(C): Decide what is the holding area altitude limit (AGL)	
1.1.2.21.4	OPERATOR(C): Decide if time of day affect the number of aircraft in it at one time	
1.1.2.21.5	OPERATOR(C): Decide if holding area permits line of sight communications	
1.1.2.21.6	OPERATOR(C): Decide if holding area permits good cover & concealment	
1.1.2.21.7	OPERATOR(C): Decide if holding area terrain is good for landings (emergency / rendezvous)	
1.1.2.21.8	OPERATOR(C): Decide what formations to use in holding area in air & on deck	
1.1.2.21.9	OPERATOR(C): Determine if on deck formations provide mutual support & 360º lookout	
1.1.2.21.10	OPERATOR(C): Decide if holding area is spun with the MACCS and GCE	
1.1.2.21.11	OPERATOR(C): Decide how will you define & disseminate hasty holding areas	
1.1.2.21.12	OPERATOR(C): Decide who has internal / external communications there	
1.1.2.21.13	OPERATOR(C): Decide what is the no communications plan	
1.1.2.21.14	OPERATOR(C): Determine if all holding areas are	

	depicted on your map	
1.1.2.22	METHOD: Formulate the FAC(A) plan	
1.1.2.22.1	OPERATOR(C): Review the ATO	X
1.1.2.22.2	OPERATOR(C): Determine the CAS routing, RW & FW	
1.1.2.22.3	OPERATOR(C): Determine what are the control points	
1.1.2.22.4	OPERATOR(C): Determine what is the air defense condition	
1.1.2.22.5	OPERATOR(C): Determine what are the air defense measures	
1.1.2.22.6	OPERATOR(C): Determine what CAS missions are during your TOS	X
1.1.2.22.7	OPERATOR(C): Determine if there will be a tanker during your TOS (call sign & freq)	
1.1.2.22.8	OPERATOR(C): Determine the SEAD SOP & the SEAD plan (regiment & battalion)	
1.1.2.22.9	OPERATOR(C): Decide who are the regimental & battalion air officers (call signs)	
1.1.2.22.10	OPERATOR(C): Decide what are the PPS / PPOC / immediate CAS missions	
1.1.2.22.11	OPERATOR(C): Decide how they apply to the fire support plan	
1.1.2.22.12	OPERATOR(C): Decide where the FACs will be	
1.1.2.22.13	OPERATOR(C): Decide what is the air officer’s plan for FAC(a) use	
1.1.2.22.14	OPERATOR(C): Decide what is the ACA description (AGL / MSL, SOP for describing hasty)	
1.1.2.22.15	OPERATOR(C): Decide what is the medevac plan	
1.1.2.22.16	OPERATOR(C): Decide what map datum will everyone be on	
1.1.2.23	METHOD: Understand the fire support plan (from the FSC)	
1.1.2.23.1	OPERATOR(C): Review groups	
1.1.2.23.2	OPERATOR(C): Review series	
1.1.2.23.3	OPERATOR(C): Review programs	
1.1.2.23.4	OPERATOR(C): Review SOP items	
1.1.2.23.5	OPERATOR(C): Determine if you have the scheduling worksheets	
1.1.2.23.6	OPERATOR(C): Determine if you have the target lists	
1.1.2.23.7	OPERATOR(C): Determine if you have the target precedence list	
1.1.2.23.8	OPERATOR(C): Determine if you have an attack guidance matrix	
1.1.2.23.9	OPERATOR(C): Decide what are all the fire support coordination measures	
1.1.2.23.10	METHOD: Decide what fire support assets are there & where will they be	
1.1.2.23.10.1	OPERATOR(C): Decide General support	
1.1.2.23.10.2	OPERATOR(C): Decide Direct support	
1.1.2.23.10.3	OPERATOR(C): Decide AN/TPQ-36/37 counter mortar & battery radar sites	
1.1.2.23.10.4	OPERATOR(C): Decide Displacement schedule	
1.1.2.23.10.5	OPERATOR(C): Decide what is the laser employment plan	

1.1.2.23.10.6	OPERATOR(C): Decide Laser spot teams	
1.1.2.23.10.7	OPERATOR(C): Decide Mule locations	
1.1.2.24	METHOD: Formulate plan for the friendly situation	
1.1.2.24.1	OPERATOR(C): Decide Higher	
1.1.2.24.2	OPERATOR(C): Decide Adjacent	
1.1.2.24.3	OPERATOR(C): Decide Support	
1.1.2.24.4	OPERATOR(C): Decide what are the maneuver control measures	
1.1.2.24.5	OPERATOR(C): Decide what is the main effort	
1.1.2.24.6	OPERATOR(C): Decide what are the reconnaissance units	
1.1.2.25	METHOD: Review all the required net frequencies	
1.1.2.25.1	OPERATOR(C): Record TATC	
1.1.2.25.2	OPERATOR(C): Record TAD	
1.1.2.25.3	OPERATOR(C): Record TAR 1 / 2	
1.1.2.25.4	OPERATOR(C): Record TACP(L)	
1.1.2.25.5	OPERATOR(C): Record COF	
1.1.2.25.6	OPERATOR(C): Record NGF spot	
1.1.2.25.7	OPERATOR(C): Record Artillery air spot	
1.1.2.25.8	OPERATOR(C): Record Regimental TAC 1 / 2	
1.1.2.25.9	OPERATOR(C): Record Battalion TAC 1 / 2	
1.1.2.25.10	OPERATOR(C): Record ADA	
1.1.2.25.11	OPERATOR(C): Record Division Air Observation net	
1.1.2.25.12	OPERATOR(C): Record MAGTF Air Observation net	
1.1.2.25.13	OPERATOR(C): Record Div / MAGTF Intelligence net	
1.1.2.25.14	OPERATOR(C): Record Div / MAGTF reconnaissance net	
1.1.2.26	METHOD: Review all the required call signs	
1.1.2.26.1	OPERATOR(C): Record TACC	
1.1.2.26.2	OPERATOR(C): Record AOC	
1.1.2.26.3	OPERATOR(C): Record DASC	
1.1.2.26.4	OPERATOR(C): Record DASC(A)	
1.1.2.26.5	OPERATOR(C): Record Regiment	
1.1.2.26.6	OPERATOR(C): Record Battalion	
1.1.2.26.7	OPERATOR(C): Record Artillery Battalion	
1.1.2.26.8	OPERATOR(C): Record Artillery Battery	
1.1.2.26.9	OPERATOR(C): Record FOS	
1.1.2.26.10	OPERATOR(C): Record FACs	
1.1.2.27	METHOD: Review the report formats	
1.1.2.27.1	OPERATOR(C): Decide what is the chattermark plan	
1.1.2.27.2	OPERATOR(C): Determine if you have authentication material	
1.1.2.27.3	OPERATOR(C): Determine the HAVEQUICK word of the day	
1.1.2.27.4	OPERATOR(C): Determine what marking capability will you have	
1.1.2.27.5	OPERATOR(C): Decide how will you use illumination	
1.1.2.28	METHOD: Develop Battle Positions	
1.1.2.28.1	OPERATOR(C): Decide if multiple battle positions required	
1.1.2.28.2	OPERATOR(C): Decide if battle positions are big enough for aircraft to maneuver in	
1.1.2.28.3	OPERATOR(C): Determine what is the AGL altitude	

	limit of the battle position	
1.1.2.28.4	OPERATOR(C): Determine if the battle position allow LOS communications	
1.1.2.28.5	OPERATOR(C): Decide if the terrain is conducive to the type of attack pattern planned	
1.1.2.28.6	OPERATOR(C): Decide if the terrain allows for cover & concealment	
1.1.2.28.7	OPERATOR(C): Decide if the Hellfire geometry is correct	
1.1.2.28.8	OPERATOR(C): Determine if you applied EOTDA data to the battle position	
1.1.2.29	METHOD: Formulate plan for visibility	
1.1.2.29.1	OPERATOR(C): Determine extent of LASER propagation	
1.1.2.29.2	OPERATOR(C): Decide if the battle position lets all aircraft support each other	
1.1.2.29.3	OPERATOR(C): Decide if all aircraft will be able to see each other	
1.1.2.29.4	OPERATOR(C): Decide if your battle positions are spun with the MACCS and the GCE	
1.1.2.29.5	OPERATOR(C): Decide how you will define & disseminate hasty battle positions	
1.1.2.30	METHOD: Formulate plan for firing points / covering points	
1.1.2.30.1	OPERATOR(C): Decide if there are multiple points for each aircraft	
1.1.2.30.2	OPERATOR(C): Decide if the aircraft are oriented to provide mutual support	
1.1.2.30.3	OPERATOR(C): Decide what communications / signals are required to shoot	
1.1.2.30.4	OPERATOR(C): Decide if the cover element see both the maneuver element and threat	
1.1.2.30.5	OPERATOR(C): Determine if firing points offer adequate (interlocking) fields of fire	
1.1.2.30.6	OPERATOR(C): Determine final attack headings	
1.1.2.30.7	OPERATOR(C): Determine if will you use tarps & rifles & have they been spun	
1.1.2.30.8	OPERATOR(C): Decide what are your weapons of choice for each firing point	
1.1.2.31	METHOD: Formulate plan for finding ranges to target	
1.1.2.31.1	OPERATOR(C): Decide how will range estimation be performed	
1.1.2.31.2	OPERATOR(C): Conduct a map study of the objective area	
1.1.2.31.3	OPERATOR(C): Decide who will LASER range (mule / NTS cobra) & report grids	
1.1.2.31.4	OPERATOR(C): Determine mil values for expected targets from battle positions	
1.1.2.31.5	OPERATOR(C): Decide what mil value correlates to out-of-range	
1.1.2.32	METHOD: Formulate plan for elevation analysis	
1.1.2.32.1	OPERATOR(C): Construct the elevation analysis diagram	
1.1.2.32.2	OPERATOR(C): Decide how various Hellfire trajectories work, given the analysis	

1.1.2.33	METHOD: Formulate plan for attack patterns	
1.1.2.33.1	METHOD: Decide who controls / coordinates the attacks:	
1.1.2.33.1.1	OPERATOR(C): Determine attack control for the section	
1.1.2.33.1.2	OPERATOR(C): Determine attack control for the division	
1.1.2.33.2	METHOD: Determine if SOP attacks been selected based on METT-TSL	
1.1.2.33.2.1	OPERATOR(C): Decide if attack altitude been based on METT-TSL	
1.1.2.33.2.2	OPERATOR(C): Decide if attack formations been based on METT-TSL	
1.1.2.33.3	OPERATOR(C): Decide if NVG attacks require tighter formations	
1.1.2.33.4	OPERATOR(C): Decide what internal & external flight communications are needed for attacks	
1.1.2.33.5	OPERATOR(C): Decide what are the single radio & no communication attack plans	
1.1.2.33.6	OPERATOR(C): Decide if multiple attacks will be run based on the threat	
1.1.2.34	METHOD: Formulate plan for target engagement	
1.1.2.34.1	OPERATOR(C): Decide what is the layout & orientation of the target	
1.1.2.34.2	OPERATOR(C): Decide if it likely to follow a doctrinal template	
1.1.2.34.3	OPERATOR(C): Decide if you use kill zones / engagement areas / boundaries	
1.1.2.34.4	METHOD: Determine the illumination plan	
1.1.2.34.4.1	OPERATOR(C): Decide who will deliver it	
1.1.2.34.4.2	OPERATOR(C): Decide Time / amount available	
1.1.2.34.4.3	OPERATOR(C): Decide Dedicated RW illumination shooters or self illumination	
1.1.2.34.4.4	OPERATOR(C): Decide if battle positions conducive to illumination delivery	
1.1.2.34.4.5	OPERATOR(C): Decide what are the plan for engaging area targets is:	
1.1.2.34.4.6	OPERATOR(C): Decide Ordnance	
1.1.2.34.4.7	OPERATOR(C): Decide Attack patterns	
1.1.2.34.4.8	OPERATOR(C): Decide Positions	
1.1.2.34.4.9	OPERATOR(C): Decide what are the best firing positions in order to maximize the beaten zone	
1.1.2.35	METHOD: Formulate plan for engaging point targets	
1.1.2.35.1	OPERATOR(C): Determine ordnance	
1.1.2.35.2	OPERATOR(C): Determine attack patterns	
1.1.2.35.3	OPERATOR(C): Determine PGM utilization	
1.1.2.35.4	OPERATOR(C): Determine positions	
1.1.2.35.5	OPERATOR(C): Determine if redundant targeting is planned for	
1.1.2.35.6	OPERATOR(C): Determine the plan for linear & lateral target engagement	
1.1.2.36	METHOD: Formulate plan for ordnance expenditure	
1.1.2.36.1	OPERATOR(C): Decide how much ordnance is required for each attack (attack matrix)	

1.1.2.37	METHOD: Formulate plan for Battle Damage Assessment	
1.1.2.37.1	OPERATOR(C): Decide how BDA reports are going to flow through the MACCS	
1.1.2.37.2	OPERATOR(C): Decide Primary net / call sign / freq	
1.1.2.37.3	OPERATOR(C): Decide Alternate net / call sign / freq	
1.1.2.38	METHOD: Formulate plan for target area tactics	
1.1.2.38.1	OPERATOR(C): Decide if surprise is desired / required	
1.1.2.38.2	OPERATOR(C): Determine if you afford to loiter in the objective area	
1.1.2.38.3	OPERATOR(C): Decide if massed firepower or continuous support required	
1.1.2.38.4	OPERATOR(C): Examine the objective area from the enemy's viewpoint	X
1.1.2.38.5	OPERATOR(C): Decide how the enemy would counter / defend against attack helicopters	
1.1.2.39	METHOD: Formulate plan for evasive maneuvers	
1.1.2.39.1	OPERATOR(C): Decide how will you use expendables (day / night / (non) effective)	
1.1.2.39.2	OPERATOR(C): Decide what calls are required inter-flight	
1.1.2.39.3	OPERATOR(C): Decide who reports the encounter (MISREP / IFREP) & when	
1.1.2.39.4	OPERATOR(C): Decide who takes the call (net / call sign / freq)	
1.1.2.40	METHOD: Formulate plan for rendezvous procedures	
1.1.2.40.1	OPERATOR(C): Decide if on deck, does the plan include mutual support and 360º lookout	
1.1.2.40.2	OPERATOR(C): Decide if the terrain is conducive to helicopter landings	
1.1.2.40.3	OPERATOR(C): Decide what altitude, airspeed, orbit direction is planned in flight	
1.1.2.40.4	OPERATOR(C): Decide what communications are required & how will you rejoin	
1.1.2.40.5	OPERATOR(C): Decide how long you will wait for the rest of the flight	
1.1.2.40.6	OPERATOR(C): Decide who will be in charge based on aircraft remaining	
1.1.2.40.7	OPERATOR(C): Decide what is the plan for go / no go at that point	
1.1.2.41	METHOD: Formulate plan for Bingo profiles	
1.1.2.41.1	OPERATOR(C): Decide how much fuel is required to complete the mission:	
1.1.2.41.2	OPERATOR(C): Decide how much fuel is required at takeoff	
1.1.2.41.3	OPERATOR(C): Decide how much fuel is required to enter the objective area	
1.1.2.41.4	OPERATOR(C): Decide how much fuel is required to egress the objective area & RTB (direct / via routing)	
1.1.2.41.5	OPERATOR(C): Determine the direct heading / fuel bingos for each checkpoint	
1.1.2.41.6	OPERATOR(C): Decide what are the ordnance / expendable bingos:	
1.1.2.41.7	OPERATOR(C): Decide how much fuel is required to	

	enter the objective area	
1.1.2.41.8	OPERATOR(C): Decide how much fuel is required to egress the objective area & RTB	
1.1.2.42	METHOD: Formulate plan for contingencies	
1.1.2.42.1	OPERATOR(C): Determine what to do in the case of overwhelming force	
1.1.2.42.2	OPERATOR(C): Determine what to do in case no enemy is detected at planned location	
1.1.2.42.3	OPERATOR(C): Determine what to do in case of no communication with the FAC or if the FAC is dead	
1.1.2.42.4	OPERATOR(C): Determine what to do in case we go Winchester prior to mission accomplishment	
1.1.2.42.5	OPERATOR(C): Determine what to do in the case aircraft go down within our flight (mechanical / enemy fire)	
1.1.3	**GOAL:** Prepare Mission Brief	
1.1.4	**GOAL:** Prepare Mission Documents	
1.1.4.1	METHOD: Prepare Maps	
1.1.4.1.1	METHOD: Determine Target Areas	
1.1.4.1.2	METHOD: Denote All FSCM to Include Friendly Positions, RFA, NFA, etc.	
1.1.4.1.3	METHOD: Determine Which CP, HA, IP, BP to Use	
1.1.4.1.4	METHOD: Draw Compass Rose Oriented Magnetic North	
1.1.4.1.5	METHOD: Draw Magnetic Radials Every 5 Degrees from IP / BP Through Target Area	
1.1.4.1.6	METHOD: Draw Scale for Nautical Miles	
1.1.4.1.7	METHOD: Draw Center of Azimuths of Fire for Known Artillery Positions	
1.1.4.1.8	METHOD: Draw Magnetic Final Attack Cones to Coordinate Run-Ins, RFA, NFA, etc.	
1.1.4.2	METHOD: Prepare Smartpack	
1.1.4.2.1	METHOD: Generate Taskview Kneeboard Card	
1.1.4.2.2	METHOD: Generate Schematic Kneeboard Card	
1.1.4.2.3	METHOD: Generate Smartpack Cover Sheet	
1.1.4.2.4	METHOD: Generate FARP or FOB Diagram	
1.1.4.2.5	METHOD: Generate Objective Area Diagram	
1.1.4.2.6	METHOD: Generate ACEOI Quick Card	
1.1.4.2.7	METHOD: Generate Pre-Spun and Blank 9-Line Cards	
1.2	**GOAL:** Brief Mission	
1.2.1	METHOD: Brief Situation	
1.2.2	METHOD: Brief Mission Statement	
1.2.3	METHOD: Brief Execution	
1.2.4	METHOD: Brief Administration	
1.2.5	METHOD: Brief Coordination & Logistics	
1.3	**GOAL:** Fly Mission	
1.3.1	**GOAL:** Prepare Mission Equipment	
1.3.1.1	METHOD: Gather Mission Essential Equipment	
1.3.1.1.1	OPERATOR(M): Take 1:250,000 scale map notated with CPs and IPs	
1.3.1.1.2	OPERATOR(M): Take 1:50,000 scale map of all planned working areas	
1.3.1.1.3	OPERATOR(M): Take target area photos	
1.3.1.1.4	OPERATOR(M): Take ATO and SPINS	

1.3.1.1.5	OPERATOR(M): Take grid reader and protractor	
1.3.1.1.6	OPERATOR(M): Take ACEOI	
1.3.1.1.7	OPERATOR(M): Take COMSEC Material Compatible with the GCE	
1.3.1.1.8	OPERATOR(M): Take Image or Gyro-Stabilized Binoculars	
1.3.1.1.9	OPERATOR(M): Take Digital Camera or Camcorder	
1.3.1.1.10	OPERATOR(M): Take Mission Playback Tapes	
1.3.1.1.11	OPERATOR(M): Take IR Pointers	
1.3.1.1.12	OPERATOR(M): Take LASER Eye Protection	
1.3.1.1.13	OPERATOR(M): Take FAC(A) Handbook	
1.3.1.1.14	OPERATOR(M): Take Kneeboard Smartpack	
1.3.1.1.15	OPERATOR(M): Take Flight Gear	
1.3.2	**GOAL:** Launch Section	
1.3.3	**GOAL:** Arrive at terminal area with full mission capability	
1.3.3.1	**GOAL:** Maximize SA Enroute	
1.3.3.1.1	METHOD: Conduct Liaison with DASC	
1.3.3.1.1.1	*SELECTION:* If LOS exists with target agency:	
1.3.3.1.1.1.1	METHOD: Conduct check-in and confirm mission parameters	
1.3.3.1.1.1.1.1	METHOD: Confirm identification	
1.3.3.1.1.1.1.1.1	*SELECTION:* If you are originating contact:	
1.3.3.1.1.1.1.1.1.1	OPERATOR(M): State call sign of contacted agency followed by call sign of aircrew ("(contacted agency call sign), this is (your call sign)")	
1.3.3.1.1.1.1.1.1.2	OPERATOR(P): Confirm contacted agency's response ("(your call sign), this is (contacted agency's call sign)")	
1.3.3.1.1.1.1.1.2	*SELECTION:* If you are being contacted:	
1.3.3.1.1.1.1.1.2.1	OPERATOR(M): Hear your call sign followed by call sign of contacting agency ("(your call sign), this is (call sign of contacting agency)")	
1.3.3.1.1.1.1.1.2.2	OPERATOR(P): Confirm contacted agency's response ("(your call sign), this is (contacted agency's call sign)")	
1.3.3.1.1.1.1.2	*SELECTION:* If working unencrypted communications:	
1.3.3.1.1.1.1.2.1	METHOD: Conduct authentication routine	
1.3.3.1.1.1.1.2.1.1	METHOD: Respond to new agency authentication query	
1.3.3.1.1.1.1.2.1.1.1	OPERATOR(P): Hear the new agency's authentication query letters	
1.3.3.1.1.1.1.2.1.1.2	OPERATOR(C): Trace the query letters on the ACEOI	
1.3.3.1.1.1.1.2.1.1.3	OPERATOR(M): Respond to agency with correct letter	
1.3.3.1.1.1.1.2.1.2	METHOD: Query new agency for authentication	
1.3.3.1.1.1.1.2.1.2.1	OPERATOR(C): Choose new query letters on the ACEOI	
1.3.3.1.1.1.1.2.1.2.2	OPERATOR(M): Query the agency with the new letters	

1.3.3.1.1.1.1.2.1.2.3	OPERATOR(P): Hear the agency's response letter	
1.3.3.1.1.1.1.2.1.2.4	OPERATOR(C): Determine correctness of agency's response	
1.3.3.1.1.1.1.2.1.3	*SELECTION:* If agency responds incorrectly first time:	
1.3.3.1.1.1.1.2.1.3.1	METHOD: Use METHOD: Query new agency for authentication	
1.3.3.1.1.1.1.2.1.4	*SELECTION:* If agency responds incorrectly second time:	
1.3.3.1.1.1.1.2.1.4.1	OPERATOR(M): Attempt agency contact on secondary frequency	
1.3.3.1.1.1.1.2.1.4.2	METHOD: Use METHOD: Conduct DASC Check-in	
1.3.3.1.1.1.1.3	METHOD: Provide mission information	
1.3.3.1.1.1.1.3.1	OPERATOR(M): State mission number per the ATO	
1.3.3.1.1.1.1.3.2	OPERATOR(M): State mission status (“Up as fragged” or “With exceptions” plus exceptions to information contained in the ATO)	
1.3.3.1.1.1.1.4	METHOD: Obtain friendly and enemy situation update	X
1.3.3.1.1.1.1.4.1	OPERATOR(P): Hear agency's situation report	
1.3.3.1.1.1.1.4.2	OPERATOR(M): Copy abbreviated report on kneeboard	
1.3.3.1.1.1.1.4.3	OPERATOR(C): Understand how information in report changes mission plan, if at all	
1.3.3.1.1.1.1.5	METHOD: Advise of FAC(A) Capability	
1.3.3.1.1.1.1.5.1	OPERATOR(M): Transmit verification of FAC(A) capability	
1.3.3.1.1.1.1.6	METHOD: Confirm Supported Unit and Unit Location	X
1.3.3.1.1.1.1.6.1	OPERATOR(C): Recall (or recheck) support unit information from ATO	
1.3.3.1.1.1.1.6.2	OPERATOR(M): Query if supported unit and location remains the same per the ATO	
1.3.3.1.1.1.2	METHOD: Request Update of CAS Missions and JTARs in Progress	
1.3.3.1.1.1.2.1	OPERATOR(C): Recall (or recheck) expected CAS mission and JTARs from ATO	
1.3.3.1.1.1.2.2	OPERATOR(M): Query whether expected CAS missions and JTARs are being executed	
1.3.3.1.1.1.3	METHOD: Obtain Routing Information	
1.3.3.1.1.1.3.1	*SELECTION:* If agency provides routing information:	
1.3.3.1.1.1.3.1.1	METHOD: Receive routing information	
1.3.3.1.1.1.3.1.1.1	OPERATOR(P): Hear agency's routing instructions	
1.3.3.1.1.1.3.1.1.2	OPERATOR(M): Copy instructions on kneeboard	
1.3.3.1.1.1.3.1.1.3	OPERATOR(M): Denote pertinent information on map	
1.3.3.1.1.1.3.1.1.4	**GOAL:** Understand implications of flying the assigned route	

1.3.3.1.1.1.3.1.1.4.1	METHOD: Determine if assigned route crosses restrictive FSCMs or known enemy positions	
1.3.3.1.1.1.3.1.1.4.1.1	OPERATOR(P): Compare route CPs to map information	
1.3.3.1.1.1.3.1.1.4.2	METHOD: Determine if assigned routing causes unacceptable changes to mission plan	
1.3.3.1.1.1.3.1.1.4.2.1	OPERATOR(C): Calculate arrival time at terminal area	
1.3.3.1.1.1.3.1.1.4.3	*SELECTION:* If assigned route interferes with successful completion of mission	
1.3.3.1.1.1.3.1.1.4.3.1	METHOD: Obtain approval of modified route	
1.3.3.1.1.1.3.1.1.4.3.1.1	OPERATOR(C): Mentally formulate reason for route rejection	
1.3.3.1.1.1.3.1.1.4.3.1.2	OPERATOR(C): Choose alternate route from accumulated information and map CP data	
1.3.3.1.1.1.3.1.1.4.3.1.3	OPERATOR(M): Explain to DASC the need for route change and offer alternative route	
1.3.3.1.1.1.3.1.1.4.3.1.4	OPERATOR(P): Hear confirmation of approval for alternate route	
1.3.3.1.1.1.3.2	*SELECTION:* If agency does not provide routing information:	
1.3.3.1.1.1.3.2.1	OPERATOR(M): Request routing information	
1.3.3.1.1.1.3.2.2	METHOD: Use METHOD: Receive routing information	
1.3.3.1.1.2	*SELECTION:* If LOS does not exist with target agency:	
1.3.3.1.1.2.1	METHOD: Attempt alternate form of contact	
1.3.3.1.1.2.1.1	OPERATOR(M): Attempt contact with agency via other airborne assets	
1.3.3.1.1.2.1.2	OPERATOR(M): Attempt contact via ground nets	
1.3.3.1.1.2.1.3	OPERATOR(M): Increase altitude within tactical limits	
1.3.3.1.1.2.2	*SELECTION:* If contact is achieved via alternate methods:	
1.3.3.1.1.2.2.1	METHOD: Use METHOD: Conduct Liaison with DASC	
1.3.3.2	**GOAL:** Navigate to terminal area	
1.3.3.2.1	OPERATOR(C): Visually match assigned route CPs to map CPs	
1.3.3.2.2	OPERATOR(C): Query copilot regarding understanding of the assigned route	
1.3.3.2.3	METHOD: Provide copilot with navigation data	
1.3.3.2.3.1	*SELECTION:* If time and workload permit:	
1.3.3.2.3.1.1	MAINTENANCE METHOD: Provide copilot with navigation updates	
1.3.3.2.3.1.1.1	*SELECTION:* If prominent terrain feature along route to next CP is visible	
1.3.3.2.3.1.1.1.1	OPERATOR(P): Identify prominent terrain feature	

1.3.3.2.3.1.1.1.2	OPERATOR(M): Advise copilot to fly toward prominent terrain feature	
1.3.3.2.3.1.1.2	*SELECTION:* If prominent terrain feature is not along route or not visible	
1.3.3.2.3.1.1.2.1	OPERATOR(C): Identify initial heading or cardinal direction to next CP	
1.3.3.2.3.1.1.2.2	OPERATOR(M): Advise copilot to fly a heading or cardinal direction	
1.3.3.2.3.2	*SELECTION:* If time and workload do not permit:	
1.3.3.2.3.2.1	METHOD: Provide copilot with digital navigation data	
1.3.3.2.3.2.1.1	OPERATOR(M): Enter route in navigation computer	
1.3.3.2.3.2.1.2	OPERATOR(M): Advise copilot that route is loaded in computer and task to fly route unassisted	
1.3.3.3	**GOAL:** Conduct Liaison with Battalion TACP	
1.3.3.3.1	*SELECTION:* If LOS with Battalion TACP exists:	
1.3.3.3.1.1	METHOD: Use METHOD: Conduct check-in and confirm mission parameters	
1.3.3.3.2	*SELECTION:* Use *SELECTION:* If LOS does not exist with target agency:	
1.3.4	**GOAL:** Maximize SA in Terminal Area	X
1.3.4.1	MAINTENANCE METHOD: Conduct continuous visual reconnaissance of working area	X
1.3.4.1.1	OPERATOR(C): Mentally Subdivide the Area	
1.3.4.1.1.1	METHOD: Search each subdivision systematically	
1.3.4.1.1.1.1	METHOD: Use available sensors to scan	
1.3.4.1.1.1.1.1	OPERATOR(P): Use binoculars	
1.3.4.1.1.1.1.2	OPERATOR(P): Use DVO / FLIR	
1.3.4.1.1.1.1.3	OPERATOR(P): Use naked eye	
1.3.4.1.1.1.2	METHOD: Look for indications of use and organization of terrain by enemy forces	
1.3.4.1.1.1.2.1	"OPERATOR(P): Note all roads, tracks, and trails"	
1.3.4.1.1.1.2.2	OPERATOR(P): Note any orderly or geometrical patterns	
1.3.4.1.1.1.2.3	OPERATOR(P): Note smoke or dust	
1.3.4.1.1.1.2.4	OPERATOR(P): Note any movement	
1.3.4.1.1.1.2.5	OPERATOR(P): Note flashes and reflections	
1.3.4.1.1.1.2.6	OPERATOR(P): Note patches of terrain lighter than surrounding area	
1.3.4.1.1.1.2.7	OPERATOR(P): Note trenches that appear clear of water (aside from desert environment)	
1.3.4.1.1.1.3	METHOD: Locate fire support units	
1.3.5	**GOAL:** Obtain terminal control	X
1.3.5.1	METHOD: Conduct Battle Handover with off-going terminal controller	X
1.3.5.1.1	METHOD: Use Battle Handover Brief format	X
1.3.5.1.1.1	METHOD: Receive Situation brief (items as applicable)	X
1.3.5.1.1.1.1	OPERATOR(P): Hear threat update	
1.3.5.1.1.1.2	OPERATOR(M): Write down or verify already have accurate threat update	
1.3.5.1.1.1.3	OPERATOR(P): Hear SAM / AAA type, location,	

	and time last active	
1.3.5.1.1.1.4	OPERATOR(M): Write down or verify already have accurate SAM / AAA type, location, and time last active	
1.3.5.1.1.1.5	OPERATOR(P): Hear threat aircraft type, location, and time sighted	
1.3.5.1.1.1.6	OPERATOR(M): Write down or verify already have accurate threat aircraft type, location, and time sighted	
1.3.5.1.1.1.7	OPERATOR(P): Hear ground forces location, time sighted, and recent BDA	
1.3.5.1.1.1.8	OPERATOR(M): Write down or verify already have accurate ground forces location, time sighted, and recent BDA	
1.3.5.1.1.1.9	OPERATOR(P): Hear friendly and supported unit update	
1.3.5.1.1.1.10	OPERATOR(M): Write down or verify already have accurate friendly and supported unit update	
1.3.5.1.1.1.11	OPERATOR(P): Hear friendly location and lead trace	
1.3.5.1.1.1.12	OPERATOR(M): Write down or verify already have accurate friendly location and lead trace	
1.3.5.1.1.1.13	OPERATOR(P): Hear listing of significant direct fires (tanks, LAV 25mm, etc.)	
1.3.5.1.1.1.14	OPERATOR(M): Write down or verify already have accurate listing of significant direct fires (tanks, LAV 25mm, etc.)	
1.3.5.1.1.1.15	OPERATOR(P): Hear description of battle space geometry (BPs, GTLs, max ordinates, etc.)	
1.3.5.1.1.1.16	OPERATOR(M): Write down or verify already have accurate description of battle space geometry (BPs, GTLs, max ordinates, etc.)	
1.3.5.1.1.1.17	OPERATOR(P): Hear list of call signs	
1.3.5.1.1.1.18	OPERATOR(M): Write down or verify already have accurate list of call signs	
1.3.5.1.1.1.19	METHOD: Receive list of FSCMs in effect (time and coordinates for each)	X
1.3.5.1.1.1.19.1	OPERATOR(P): Hear FSCL / CFL / BCL / RFL	
1.3.5.1.1.1.19.2	OPERATOR(M): Write down or verify already have accurate FSCL / CFL / BCL / RFL	
1.3.5.1.1.1.19.3	OPERATOR(P): Hear RFA / NFA / FFA	
1.3.5.1.1.1.19.4	OPERATOR(M): Write down or verify already have accurate RFA / NFA / FFA	
1.3.5.1.1.1.19.5	OPERATOR(P): Hear ACA	
1.3.5.1.1.1.19.6	OPERATOR(M): Write down or verify already have accurate ACA	
1.3.5.1.1.1.19.7	OPERATOR(P): Hear phase lines	
1.3.5.1.1.1.19.8	OPERATOR(M): Write down or verify already have accurate phase lines	
1.3.5.1.1.1.19.9	OPERATOR(P): Hear boundaries	
1.3.5.1.1.1.19.10	OPERATOR(M): Write down or verify already have accurate boundaries	
1.3.5.1.1.1.19.11	OPERATOR(P): Hear area reference system	
1.3.5.1.1.1.19.12	OPERATOR(M) Write down or verify already have area accurate reference system	

1.3.5.1.1.2	METHOD: Receive Mission Brief	X
1.3.5.1.1.2.1	OPERATOR(P): Hear list of air missions in progress	
1.3.5.1.1.2.2	OPERATOR(M): Write down or verify already have accurate list of air missions in progress	
1.3.5.1.1.2.3	OPERATOR(P): Hear list of air missions expected	
1.3.5.1.1.2.4	OPERATOR(M): Write down or verify already have accurate list of air missions expected	
1.3.5.1.1.2.5	OPERATOR(P): Hear list of indirect fire missions in progress	
1.3.5.1.1.2.6	OPERATOR(M): Write down or verify already have accurate list of indirect fire missions in progress	
1.3.5.1.1.2.7	OPERATOR(P): Hear list of indirect fire missions expected	
1.3.5.1.1.2.8	OPERATOR(M): Write down or verify already have accurate list of indirect fire missions expected	
1.3.5.1.1.3	METHOD: Receive Execution Brief	X
1.3.5.1.1.3.1	METHOD: Receive list of aircraft on station	
1.3.5.1.1.3.1.1	OPERATOR(P): Hear mission numbers	
1.3.5.1.1.3.1.2	OPERATOR(M): Write down or verify already have accurate mission numbers	
1.3.5.1.1.3.1.3	OPERATOR(P): Hear call signs	
1.3.5.1.1.3.1.4	OPERATOR(M): Write down or verify already accurate have call signs	
1.3.5.1.1.3.1.5	OPERATOR(P): Hear numbers and types with exceptions from the ATO	
1.3.5.1.1.3.1.6	OPERATOR(M): Write down or verify already have accurate numbers and types with exceptions from the ATO	
1.3.5.1.1.3.1.7	OPERATOR(P): Hear list of ordnance	
1.3.5.1.1.3.1.8	OPERATOR(M): Write down or verify already accurate have list of ordnance	
1.3.5.1.1.3.1.9	OPERATOR(P): Hear locations and altitudes	
1.3.5.1.1.3.1.10	OPERATOR(M): Write down or verify already have accurate locations and altitudes	
1.3.5.1.1.3.1.11	OPERATOR(P): Hear times on station remaining	
1.3.5.1.1.3.1.12	OPERATOR(M): Write down or verify already have accurate times on station remaining	
1.3.5.1.1.3.1.13	OPERATOR(P): Hear frequencies	
1.3.5.1.1.3.1.14	OPERATOR(M): Write down or verify already have accurate frequencies	
1.3.5.1.1.3.1.15	OPERATOR(P): Hear call sign of terminal attack controller for each aircraft or JTAR it supported	
1.3.5.1.1.3.1.16	OPERATOR(M): Write down or verify already have accurate call sign of terminal attack controller for each aircraft or JTAR it supported	
1.3.5.1.1.4	METHOD: Receive Administration and Logistics Brief	
1.3.5.1.1.4.1	METHOD: Receive list of active JTARs	
1.3.5.1.1.4.1.1	OPERATOR(P): Hear request number and time submitted	
1.3.5.1.1.4.1.1.1	OPERATOR(M): Write down or verify already have accurate request number and time submitted	
1.3.5.1.1.4.1.1.2	OPERATOR(P): Hear call sign of terminal attack controller	
1.3.5.1.1.4.1.1.3	OPERATOR(C): Write down or verify already	

	have accurate call sign of terminal attack controller	
1.3.5.1.1.4.1.1.4	OPERATOR(P): Hear CAS brief	
1.3.5.1.1.4.1.1.5	OPERATOR(M): Write down or verify already have accurate CAS brief	
1.3.5.1.1.4.1.2	METHOD: Receive list of active ASRs and type (CASEVAC, re-supply, etc.)	
1.3.5.1.1.4.1.2.1	OPERATOR(P): Hear request number and time submitted	
1.3.5.1.1.4.1.2.2	OPERATOR(M): Write down or verify already have accurate request number and time submitted	
1.3.5.1.1.4.1.2.3	OPERATOR(P): Hear supported unit	
1.3.5.1.1.4.1.2.4	OPERATOR(M): Write down or verify already have accurate name of supported unit	
1.3.5.1.1.4.1.2.5	OPERATOR(P): Hear location	
1.3.5.1.1.4.1.2.6	OPERATOR(M): Write down or verify already have accurate location	
1.3.5.1.1.5	METHOD: Receive Command and Signal Brief	
1.3.5.1.1.5.1	METHOD: Receive list of FAC(A)s on station	
1.3.5.1.1.5.1.1	OPERATOR(P): Hear call signs	
1.3.5.1.1.5.1.2	OPERATOR(M): Write down or verify already have accurate call signs	
1.3.5.1.1.5.1.3	OPERATOR(P): Hear frequencies	
1.3.5.1.1.5.1.4	OPERATOR(M): Write down or verify already have accurate frequencies	
1.3.5.1.1.5.1.5	OPERATOR(P): Hear locations	
1.3.5.1.1.5.1.6	OPERATOR(M): Write down or verify already have accurate locations	
1.3.5.1.1.6	METHOD: Receive recommendations from off-going FAC(A)	X
1.3.5.1.1.6.1	OPERATOR(M): Write down recommendations	
1.3.5.1.1.6.2	OPERATOR(C): Understand recommendations	
1.3.5.1.2	METHOD: Prepare for terminal control	X
1.3.5.1.2.1	OPERATOR(C): Understand all details and implications of the Battle Handover brief	
1.3.5.1.2.2	OPERATOR(C): Understand the GCE commander's targeting priorities	
1.3.5.1.2.3	OPERATOR(C): Understand the GCE commander's intent	
1.3.5.1.2.4	OPERATOR(C): Understand how to accomplish the GCE commander's intent	
1.3.5.1.2.5	METHOD: Evaluate preplanned mission tools	
1.3.5.1.2.5.1	OPERATOR(C): Understand how new information changes preplanned 9-lines, holding locations, and supporting aircraft tactics, if at all	
1.3.5.1.2.5.2	OPERATOR(M): Make any required adjustments to preplanned mission tools	
1.3.5.1.3	METHOD: Accept terminal control	
1.3.5.1.3.1	OPERATOR(M): State readiness to assume control (“(your call sign) is ready to accept terminal control.”)	
1.3.5.1.3.2	OPERATOR(P): Hear off-going terminal controller pass control (“(your call sign) has terminal control.”)	
1.3.5	**GOAL:** Achieve desired effects on enemy forces	X
1.3.5.1	**GOAL:** Manage FW Attacks	X
1.3.5.1.1	METHOD: Confirm terminal procedures	

1.3.5.1.1.1	METHOD: Brief general plan to AO, JTAC, or supported unit commander	
1.3.5.1.1.2	METHOD: Confirm who provides target mark	
1.3.5.1.1.3	METHOD: Confirm who provides terminal attack control	
1.3.5.1.1.4	METHOD: Confirm when attack will take place	
1.3.5.1.1.5	METHOD: Confirm that you have final approval to run the mission	
1.3.5.1.2	METHOD: Conduct JCAS check-in brief with supporting aircraft	X
1.3.5.1.2.1	METHOD: Use METHOD: Confirm identification	
1.3.5.1.2.2	*SELECTION:* Use *SELECTION:* If working unencrypted communications:	
1.3.5.1.2.3	OPERATOR(C): Reference preplanned holding IP	
1.3.5.1.2.4	*SELECTION:* If immediate (within 30 minutes) use of supporting aircraft is not predicted:	
1.3.5.1.2.4.1	*SELECTION:* If tanker is on station:	
1.3.5.1.2.4.1.1	OPERATOR(M): Instruct supporting aircraft to fill tanks, then route to IP to follow	
1.3.5.1.2.4.1.2	METHOD: Issue holding instructions	X
1.3.5.1.2.4.1.2.1	OPERATOR(M): Pass holding altitude and location by codeword (“Hold at Chevy, angels base plus 15.”)	
1.3.5.1.2.4.1.2.2	*SELECTION:* If other friendly aircraft are transiting or holding in vicinity of chosen IP:	
1.3.5.1.2.4.1.2.2.1	OPERATOR(M): Brief other aircraft information (“Eyes out for (other aircraft call sign), a section of (aircraft type), is holding at (location), at (altitude).”	
1.3.5.1.2.4.2	*SELECTION:* If tanker is not on station:	
1.3.5.1.2.4.2.1	*SELECTION:* If other available aircraft have a longer time on station:	
1.3.5.1.2.4.2.1.1	OPERATOR(M): Instruct supporting aircraft to contact DASC to find other work	
1.3.5.1.2.4.2.2	*SELECTION:* If other available aircraft have a shorter time on station:	
1.3.5.1.2.4.2.2.1	OPERATOR(M): Use METHOD: Issue holding instructions	
1.3.5.1.2.4.2.2.2	OPERATOR(M): Instruct other available aircraft to contact DASC to find other work	
1.3.5.1.2.5	METHOD: Confirm mission information	
1.3.5.1.2.5.1	OPERATOR(P): Hear mission number per the ATO	
1.3.5.1.2.5.2	OPERATOR(M): Write down mission number per the ATO	
1.3.5.1.2.5.3	OPERATOR(P): Hear mission status (“Up as fragged” or “With exceptions” plus exceptions to information contained in the ATO)	
1.3.5.1.2.5.4	OPERATOR(M): Write down mission information	
1.3.5.1.2.5.5	OPERATOR(P): Hear number and type of aircraft	
1.3.5.1.2.5.6	OPERATOR(M): Write down number and type of aircraft	
1.3.5.1.2.5.7	OPERATOR(P): Hear position and altitude	
1.3.5.1.2.5.8	OPERATOR(M): Write down position and altitude	

1.3.5.1.2.5.9	OPERATOR(P): Hear list of ordnance	
1.3.5.1.2.5.10	OPERATOR(M): Write down ordnance	
1.3.5.1.2.5.11	OPERATOR(P): Hear remaining time on station	
1.3.5.1.2.5.12	OPERATOR(M): Write down remaining time on station and current time	
1.3.5.1.2.5.13	OPERATOR(M): Pass abort code	
1.3.5.1.2.5.14	OPERATOR(M): Pass CAS aircraft laser codes in use	
1.3.5.1.2.5.15	METHOD: Confirm Aircraft has GPS Time	
1.3.5.1.2.5.15.1	OPERATOR(M): State query: "Understand using GPS time?"	
1.3.5.1.2.5.15.2	*SELECTION:* If agency response is negative:	
1.3.5.1.2.5.15.2.1	OPERATOR(M): State "Stand by for time hack."	
1.3.5.1.2.5.15.2.2	OPERATOR(P): Hear agency reply "Ready for time hack"	
1.3.5.1.2.5.15.2.3	OPERATOR(P): Observe number of seconds until top of the next minute	
1.3.5.1.2.5.15.2.4	OPERATOR(M): State number of seconds until top of the next minute (example: "Time 1415 in 20 seconds.")	
1.3.5.1.2.5.15.2.5	OPERATOR(M): At ten seconds prior to the minute, provide one-second announcements of time	
1.3.5.1.2.5.15.2.6	OPERATOR(M): At the even minute, announce "Hack."	
1.3.5.1.3	METHOD: Use METHOD: Evaluate preplanned mission tools	
1.3.5.1.4	METHOD: Build Supporting Aircraft SA	X
1.3.5.1.4.1	METHOD: Use METHOD: Issue Holding Instructions	
1.3.5.1.4.2	METHOD: Issue Attack Brief	
1.3.5.1.4.2.1	OPERATOR(M): Brief friendly situation	
1.3.5.1.4.2.2	OPERATOR(M): Brief enemy situation	
1.3.5.1.4.2.3	OPERATOR(M): Describe target area	
1.3.5.1.4.2.4	OPERATOR(M): Pass known friendly and enemy positions	
1.3.5.1.4.2.5	OPERATOR(M): Pass other airborne assets on station	
1.3.5.1.4.2.6	OPERATOR(M): Pass last target attacked	
1.3.5.1.4.2.7	OPERATOR(M): Brief expected suppression missions	
1.3.5.1.4.2.8	OPERATOR(M): Brief target area coordination (FW, RW, Other Supporting Arms, Maneuver Units)	
1.3.5.1.4.2.9	METHOD: Provide CAS Brief	X
1.3.5.1.4.2.9.1	METHOD: Issue 9-Line	X
1.3.5.1.4.2.9.1.1	OPERATOR(P): Reference preplanned 9-line	
1.3.5.1.4.2.9.1.2	OPERATOR(M): Pass IP	
1.3.5.1.4.2.9.1.3	OPERATOR(M): Pass magnetic attack heading and offset	
1.3.5.1.4.2.9.1.4	OPERATOR(M): Pass distance from IP to target	
1.3.5.1.4.2.9.1.5	OPERATOR(M): Pass target elevation in feet	

	MSL	
1.3.5.1.4.2.9.1.6	OPERATOR(M): Pass target description and activity	
1.3.5.1.4.2.9.1.7	OPERATOR(M): Pass target location in grid format	
1.3.5.1.4.2.9.1.8	*SELECTION:* If using laser mark:	
1.3.5.1.4.2.9.1.8.1	OPERATOR(M): say "LASER" followed by laser designator code	
1.3.5.1.4.2.10	METHOD: Issue Additional Remarks	X
1.3.5.1.4.2.10.1	METHOD: Evaluate threat	X
1.3.5.1.4.2.10.1.1	*SELECTION:* If enemy forces have integrated ADA (day or night), or non-integrated ADA (day):	
1.3.5.1.4.2.10.1.1.1	OPERATOR(M): Pass high threat with brief description of evaluation	
1.3.5.1.4.2.10.1.2	*SELECTION:* If enemy forces have non-integrated ADA (night):	
1.3.5.1.4.2.10.1.2.1	OPERATOR(M): Pass medium threat with brief description of evaluation	
1.3.5.1.4.2.10.1.3	*SELECTION:* If enemy forces have no ADA:	
1.3.5.1.4.2.10.1.3.1	OPERATOR(M): Pass low threat with brief description of evaluation	
1.3.5.1.4.2.10.2	METHOD: Brief SEAD During CAS Attack	X
1.3.5.1.4.2.10.3	METHOD: Brief Illumination During CAS Attack	
1.3.5.1.4.2.10.4	*SELECTION:* If laser used as mark	
	METHOD: Brief Laser to Target Line	
1.3.5.1.4.2.10.5	METHOD: Brief Gun Target Line	
1.3.5.1.4.2.10.6	METHOD: Brief Hazards to Flight	
1.3.5.1.4.2.10.7	METHOD: Brief Weather in Target Area	
1.3.5.1.4.2.10.8	METHOD: Brief Number of Weapons to Expend	
1.3.5.1.4.2.10.9	METHOD: Brief Final Attack Heading or final Attack Cone	
1.3.5.1.4.2.10.10	do math on laser geometry	
1.3.5.1.4.2.10.11	METHOD: Brief ACAs	
1.3.5.1.4.2.10.12	METHOD: Brief if Danger Close (with Commander's Initials)	
1.3.5.1.4.2.10.13	METHOD: Brief any Additional Target Information	
1.3.5.1.4.2.10.14	METHOD: Brief any Restrictions	
1.3.5.1.4.2.10.15	OPERATOR(M): Issue TOT or TTT	
1.3.5.1.4.2.11	METHOD: Confirm Receipt of CAS Brief	
1.3.5.1.4.2.11.1	*SELECTION:* If portion of brief repeat is requested:	
1.3.5.1.4.2.11.1.1	OPERATOR(M): Repeat that portion	
1.3.5.1.4.2.11.1.2	OPERATOR(P): Hear support aircraft read back target elevation, location, and any restrictions.	
1.3.5.1.4.2.11.1.3	OPERATOR(P): Hear support aircraft read back TOT or TTT	
1.3.5.1.4.2.11.2	*SELECTION:* if any portion of brief is needs to be changed:	
1.3.5.1.4.2.11.2.1	OPERATOR(M): Using the codeword for	

	"change" as a preface, say the line numbers for change followed by the new information	
1.3.5.1.5	METHOD: Conduct attack	X
1.3.5.1.5.1	METHOD: Aurally acquire support aircraft	X
1.3.5.1.5.1.1	OPERATOR(P): Hear IP inbound call ("(call sign) (IP name) inbound")	
1.3.5.1.5.1.2	OPERATOR(P): Scan target area	
1.3.5.1.5.1.3	OPERATOR(C): Choose prominent terrain near target likely to be visible from support aircraft viewpoint	
1.3.5.1.5.2	METHOD: Visually acquire support aircraft	X
1.3.5.1.5.2.1	OPERATOR(P): See IP on map	
1.3.5.1.5.2.2	OPERATOR(P): See your location on map	
1.3.5.1.5.2.3	OPERATOR(C): Determine azimuth from which support aircraft is likely to appear	
1.3.5.1.5.2.4	METHOD: (Does not preclude continuation of follow-on methods): Visually scan appropriate azimuth for support aircraft	
1.3.5.1.5.2.4.1	*SELECTION:* If support aircraft is in visual range:	
1.3.5.1.5.2.4.1.1	OPERATOR(M): Report "Visual"	
1.3.5.1.5.2.4.1.2	METHOD: Provide further talk-on	
1.3.5.1.5.2.4.2	*SELECTION:* If support aircraft is not in visual range:	
1.3.5.1.5.2.4.2.1	OPERATOR(M): Report "Continue"	
1.3.5.1.5.2.4.2.2	METHOD: Use METHOD: Visually scan appropriate azimuth for support aircraft	
1.3.5.1.5.2.5	METHOD: (Does not preclude follow-on methods): Provide talk-on	X
1.3.5.1.5.2.5.1	METHOD: Use visual "funnel" for support aircraft talk-on	X
1.3.5.1.5.2.5.1.1	OPERATOR(M): Query if support aircraft sees largest feature in target area ("Do you see the ridgeline running north-south?")	
1.3.5.1.5.2.5.1.2	OPERATOR(P): Hear support aircraft response	
1.3.5.1.5.2.5.1.3	*SELECTION:* If you are not confident that support aircraft is looking at the correct feature	
1.3.5.1.5.2.5.1.3.1	OPERATOR(M): Ask support aircraft to describe feature	
1.3.5.1.5.2.5.1.3.2	*SELECTION:* If support aircraft description is not accurate:	
1.3.5.1.5.2.5.1.3.2.1	OPERATOR(P): Choose other prominent feature near target likely to be visible from support aircraft viewpoint	
1.3.5.1.5.2.5.1.3.2.2	METHOD: Use METHOD: Use visual "funnel" for support aircraft talk-on	X
1.3.5.1.5.2.5.2	*SELECTION:* If support aircraft does not see feature:	
1.3.5.1.5.2.5.2.1	OPERATOR(P): Choose other prominent feature near target likely to be visible from support aircraft viewpoint	
1.3.5.1.5.2.5.2.2	METHOD: Use METHOD: Use visual "funnel" for support aircraft talk-on	X
1.3.5.1.5.2.5.3	*SELECTION:* If support aircraft sees feature and	

	feature is not target:	
1.3.5.1.5.2.5.3.1	OPERATOR(P): Choose smaller prominent feature closer to target that is likely to be visible from support aircraft viewpoint	
1.3.5.1.5.2.5.3.2	METHOD: Use METHOD: Use visual "funnel" for support aircraft talk-on	X
1.3.5.1.5.2.5.4	*SELECTION:* If support aircraft sees feature and feature is actual target:	
1.3.5.1.5.2.5.4.1	OPERATOR(M): Ask support aircraft to describe target	
1.3.5.1.5.2.5.4.2	*SELECTION:* If you are not confident that support aircraft is looking at the correct feature	
1.3.5.1.5.2.5.4.2.1	OPERATOR(M): Ask support aircraft to describe feature	
1.3.5.1.5.2.5.4.2.2	*SELECTION:* If support aircraft description is not accurate:	
1.3.5.1.5.2.5.4.2.2.1	OPERATOR(P): Choose other prominent feature near target likely to be visible from support aircraft viewpoint	
1.3.5.1.5.2.5.4.2.2.2	METHOD: Use METHOD: Use visual "funnel" for support aircraft talk-on	X
1.3.5.1.5.3	*SELECTION:* If support aircraft reports "In" ("(call sign) is in"):	
1.3.5.1.5.3.1	*SELECTION:* If support aircraft is in visual range:	
1.3.5.1.5.3.1.1	OPERATOR(M): Report "Visual"	
1.3.5.1.5.3.2	*SELECTION:* If support aircraft is not in visual range:	
1.3.5.1.5.3.2.1	OPERATOR(M): Report "Continue"	
1.3.5.1.5.3.3	METHOD: Provide Target Mark	X
1.3.5.1.5.3.3.1	*SELECTION:* If using other than LASER for the mark, or using in addition to LASER, and your section provides the mark:	
1.3.5.1.5.3.3.1.1	METHOD: Calculate marking ordnance release time	
1.3.5.1.5.3.3.1.1.1	OPERATOR(C): Determine distance from your section to target	
1.3.5.1.5.3.3.1.1.2	OPERATOR(C): Determine time of flight for marking ordnance	
1.3.5.1.5.3.3.1.1.3	OPERATOR(C): Add time of flight plus 30 seconds	
1.3.5.1.5.3.3.1.2	OPERATOR(M): Release marking ordnance at target at time determined by calculation	
1.3.5.1.5.3.3.1.3	METHOD: Verify support aircraft has the mark	X
1.3.5.1.5.3.3.1.3.1	OPERATOR(M): Query support aircraft "Do you have the mark?"	
1.3.5.1.5.3.3.1.3.2	OPERATOR(P): Hear support aircraft response	
1.3.5.1.5.3.3.1.3.3	*SELECTION:* If support aircraft response is negative:	
1.3.5.1.5.3.3.1.3.3.1	METHOD: Use METHOD: (Does not preclude follow-on methods): Provide talk-on	
1.3.5.1.5.3.3.2	*SELECTION:* If using LASER for marking:	
1.3.5.1.5.3.3.2.1	OPERATOR(P): Hear support aircraft call	

	"Ten seconds"	
1.3.5.1.5.3.3.2.2	OPERATOR(M): Report "Roger, ten seconds"	
1.3.5.1.5.3.3.2.3	OPERATOR(C): Prepare to fire LASER in ten seconds	
1.3.5.1.5.3.3.2.4	OPERATOR(P): Hear support aircraft call "LASER on"	
1.3.5.1.5.3.3.2.5	OPERATOR(M): Fire LASER at ten seconds past support aircraft "ten seconds" call regardless of whether support aircraft made "LASER on" call	
1.3.5.1.5.3.3.2.6	OPERATOR(M): Report "LASER on"	
1.3.5.1.5.3.3.2.7	*SELECTION:* If support aircraft reports "Spot"	
1.3.5.1.5.3.3.2.7.1	METHOD: Use METHOD: Clear support aircraft for ordnance release	
1.3.5.1.5.3.3.2.8	*SELECTION:* If support aircraft reports "Negative LASER"	
1.3.5.1.5.3.3.2.8.1	REPAIR METHOD: Command other aircraft in section to fire LASER at the target	
1.3.5.1.5.3.4	METHOD: Clear support aircraft for ordnance release	X
1.3.5.1.5.3.4.1	OPERATOR(P): Hear support aircraft call "Wings level"	
1.3.5.1.5.3.4.2	METHOD: Verify support aircraft within constraints	
1.3.5.1.5.3.4.2.1	OPERATOR(P): Determine if support aircraft nose is pointed at the target	
1.3.5.1.5.3.4.2.2	OPERATOR(P): Determine if direct line from support aircraft nose to target intersects any friendly positions	
1.3.5.1.5.3.4.2.3	OPERATOR(C): Based on prior talk-on, determine if support aircraft has the target in sight	
1.3.5.1.5.3.4.3	*SELECTION:* If support aircraft is within constraints:	
1.3.5.1.5.3.4.3.1	OPERATOR(M): Inform support aircraft "Cleared hot"	X
1.3.5.1.5.3.4.4	*SELECTION:* If support aircraft is not within constraints:	
1.3.5.1.5.3.4.4.1	OPERATOR(M): Inform support aircraft "Abort"	X
1.3.5.1.5.3.5	*SELECTION:* If LASER was used for the mark:	
1.3.5.1.5.3.5.1	*SELECTION:* If support aircraft ordnance has impacted, or support aircraft calls "Terminate":	
1.3.5.1.5.3.5.2	OPERATOR(M): Stop firing the LASER	
1.3.5.1.5.3.5.3	OPERATOR(M): Report to support aircraft "Roger, terminate"	
1.3.5.2	**GOAL:** Egress the support aircraft from the target area	
1.3.5.2.1	OPERATOR(M): Inform support aircraft attack is complete	
1.3.5.2.2	OPERATOR(M): Instruct support aircraft to egress the target area per 9-line	
1.3.5.2.3	OPERATOR(M): Pass distance of closest friendly forces to target	

1.3.5.2.4	OPERATOR(M): Pass egress instructions	
1.3.5.2.5	OPERATOR(M): Instruct support aircraft to stand by for Battle Damage Assessment	
1.3.6	**GOAL:** Evaluate Attack Damage	X
1.3.6.1	METHOD: Record attack damage	X
1.3.6.1.1	OPERATOR(P): Scan target area	
1.3.6.1.2	OPERATOR(C): Determine extent of damage	
1.3.6.1.3	OPERATOR(M): Write down extent of damage	
1.3.6.2	METHOD: Brief support aircraft using Battle Damage Assessment format	X
1.3.6.2.1	OPERATORI(M): Report size of enemy unit damaged	
1.3.6.2.2	OPERATOR(M): Report location of enemy unit	
1.3.6.2.3	OPERATOR(M): Report activity of enemy unit at time of attack	
1.3.6.2.4	OPERATOR(M): Report time of attack	
1.3.6.2.5	OPERATOR(M): Report observed damage to enemy unit	
1.3.6.3	**GOAL:** Ensure continuity of terminal control	X
1.3.6.3.1	METHOD: No later than Joker fuel state, notify AO and DASC of remaining time on station	
1.3.6.3.2	METHOD: Prepare Battle Handover brief	X
1.3.6.3.3	METHOD: Conduct Battle Handover with on-coming terminal controller	X
1.3.6.3.3.1	METHOD: Use Battle Handover Brief format	X
1.3.6.3.3.1.1	METHOD: Brief Situation (items as applicable)	X
1.3.6.3.3.1.1.1	OPERATOR(M): Pass threat update	
1.3.6.3.3.1.1.2	OPERATOR(M): Pass SAM / AAA type, location, and time last active	
1.3.6.3.3.1.1.3	OPERATOR(M): Pass threat aircraft type, location, and time sighted	
1.3.6.3.3.1.1.4	OPERATOR(M): Pass ground forces location, time sighted, and recent BDA	
1.3.6.3.3.1.1.5	OPERATOR(M): Pass friendly and supported unit update	
1.3.6.3.3.1.1.6	OPERATOR(M): Pass friendly location and lead trace	
1.3.6.3.3.1.1.7	OPERATOR(M): Pass listing of significant direct fires (tanks, LAV 25mm, etc.)	
1.3.6.3.3.1.1.8	OPERATOR(M): Pass description of battle space geometry (BPs, GTLs, max ordinates, etc.)	
1.3.6.3.3.1.1.9	OPERATOR(M): Pass list of call signs	
1.3.6.3.3.1.1.10	METHOD: Brief list of FSCMs in effect (time and coordinates for each)	X
1.3.6.3.3.1.1.10.1	OPERATOR(M): Pass FSCL / CFL / BCL / RFL	
1.3.6.3.3.1.1.10.2	OPERATOR(M): Pass RFA / NFA / FFA	
1.3.6.3.3.1.1.10.3	OPERATOR(M): Pass ACA	
1.3.6.3.3.1.1.10.4	OPERATOR(M): Pass phase lines	
1.3.6.3.3.1.1.10.5	OPERATOR(M): Pass boundaries	
1.3.6.3.3.1.1.11	OPERATOR(M) Pass area reference system	
1.3.6.3.3.1.2	METHOD: Brief Mission	X
1.3.6.3.3.1.2.1	OPERATOR(M): Pass air missions in progress	
1.3.6.3.3.1.2.2	OPERATOR(M): Pass air missions expected	
1.3.6.3.3.1.2.3	OPERATOR(M): Pass indirect fire missions in	

	progress	
1.3.6.3.3.1.2.4	OPERATOR(M): Pass indirect fire missions expected	
1.3.6.3.3.1.3	METHOD: Brief Execution	X
1.3.6.3.3.1.3.1	METHOD: Provide list of aircraft on station	
1.3.6.3.3.1.3.1.1	OPERATOR(M): Pass mission numbers	
1.3.6.3.3.1.3.1.2	OPERATOR(M): Pass call signs	
1.3.6.3.3.1.3.1.3	OPERATOR(M): Pass numbers and types with exceptions from the ATO	
1.3.6.3.3.1.3.1.4	OPERATOR(M): Pass list of ordnance	
1.3.6.3.3.1.3.1.5	OPERATOR(M): Pass locations and altitudes	
1.3.6.3.3.1.3.1.6	OPERATOR(M): Pass times on station remaining	
1.3.6.3.3.1.3.1.7	OPERATOR(M): Pass frequencies	
1.3.6.3.3.1.3.1.8	OPERATOR(M): Pass call sign of terminal attack controller for each aircraft or JTAR it supported	
1.3.6.3.3.1.4	METHOD: Brief Administration and Logistics	
1.3.6.3.3.1.4.1	METHOD: Provide list of active JTARs	
1.3.6.3.3.1.4.1.1	OPERATOR(M): Pass request number and time submitted	
1.3.6.3.3.1.4.1.2	OPERATOR(M): Pass JTAR terminal controller call signs	
1.3.6.3.3.1.4.1.3	OPERATOR(M): Pass CAS brief	
1.3.6.3.3.1.4.2	METHOD: Provide list of active ASRs and type (CASEVAC, re-supply, etc.)	
1.3.6.3.3.1.4.2.1	OPERATOR(M): Pass request number and time submitted	
1.3.6.3.3.1.4.2.2	OPERATOR(M): Pass name of supported unit	
1.3.6.3.3.1.4.2.3	OPERATOR(M): Pass location	
1.3.6.3.3.1.5	METHOD: Brief Command and Signal	
1.3.6.3.3.1.5.1	METHOD: Receive list of FAC(A)s on station	
1.3.6.3.3.1.5.1.1	OPERATOR(M): Pass call signs	
1.3.6.3.3.1.5.1.2	OPERATOR(M): Pass frequencies	
1.3.6.3.3.1.5.1.3	OPERATOR(M): Pass locations	
1.3.6.3.3.1.6.1.4	METHOD: Provide recommendations to on-coming FAC(A)	X
1.3.6.3.3.1.6.1	OPERATOR(C): Recall any pertinent information not covered by the Battle Handover brief	
1.3.6.3.3.1.6.2	OPERATOR(M): Pass pertinent information	
1.3.6.3.3.2	METHOD: Pass terminal control	X
1.3.6.3.3.2.1	OPERATOR(P): Hear on-coming controller request control by the phrase "(on-coming controller call sign) is ready to accept terminal control."	
1.3.6.3.3.2.2	OPERATOR(M): State that on-coming terminal controller has terminal control ("(on-coming controller call sign) has terminal control.")	
1.3.7	**GOAL:** Arrive at FOB or FARP with full mission capability	
1.3.7.1	**GOAL:** Maximize SA Enroute	
1.3.7.1.1	METHOD: Conduct Liaison with DASC	
1.3.7.1.1.1	*SELECTION:* If LOS exists with target agency:	

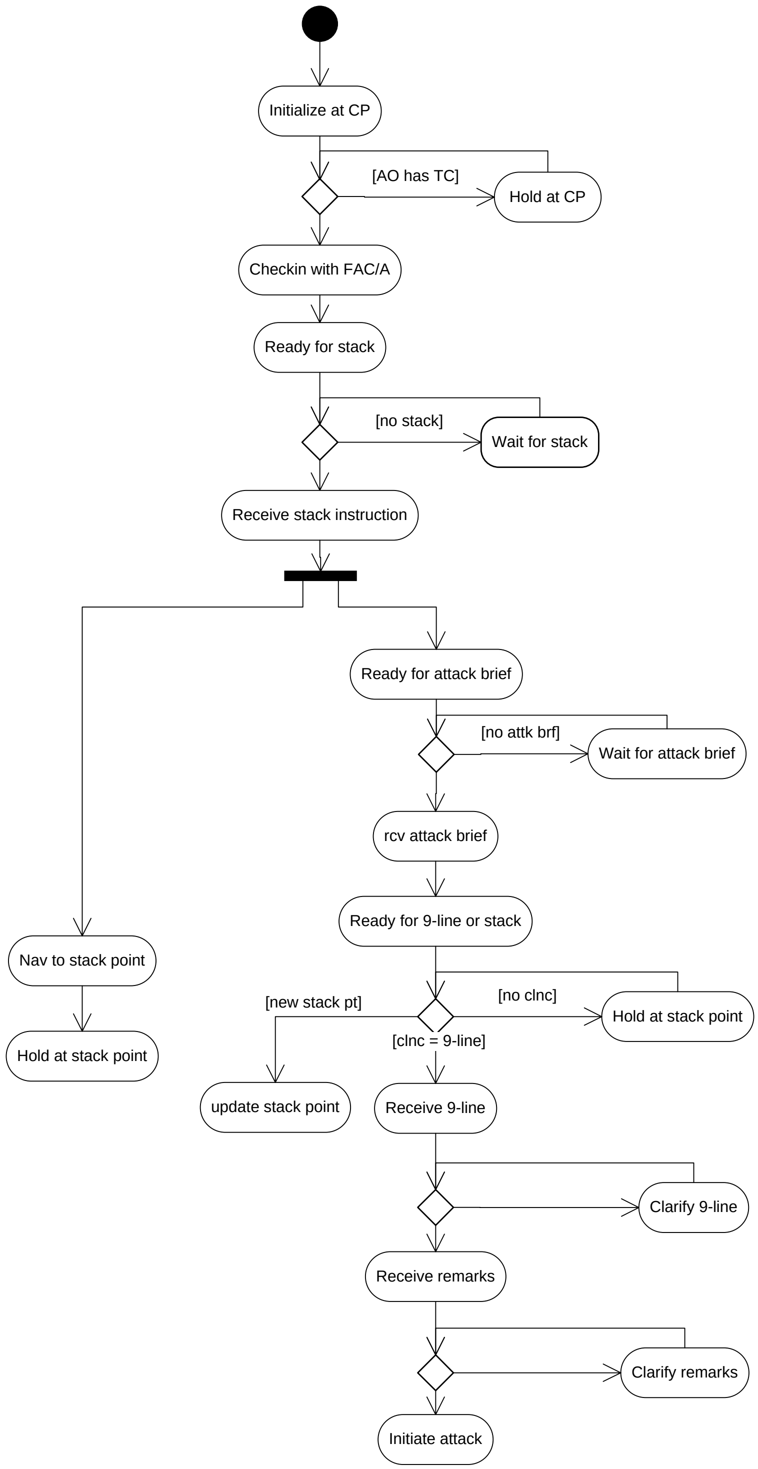
1.3.7.1.1.1.1	METHOD: Conduct check-in and confirm mission parameters	
1.3.7.1.1.1.1.1	METHOD: Confirm identification	
1.3.7.1.1.1.1.1.1	*SELECTION:* If you are originating contact:	
1.3.7.1.1.1.1.1.1.1	OPERATOR(M): State call sign of contacted agency followed by call sign of aircrew (“(contacted agency call sign), this is (your call sign)”)	
1.3.7.1.1.1.1.1.1.2	OPERATOR(P): Confirm contacted agency's response (“(your call sign), this is (contacted agency's call sign)”)	
1.3.7.1.1.1.1.1.2	*SELECTION:* If you are being contacted:	
1.3.7.1.1.1.1.1.2.1	OPERATOR(M): Hear your call sign followed by call sign of contacting agency (“(your call sign), this is (call sign of contacting agency)”)	
1.3.7.1.1.1.1.1.2.2	OPERATOR(P): Confirm contacted agency's response (“(your call sign), this is (contacted agency's call sign)”)	
1.3.7.1.1.1.1.2	*SELECTION:* If working unencrypted communications:	
1.3.7.1.1.1.1.2.1	METHOD: Conduct authentication routine	
1.3.7.1.1.1.1.2.1.1	METHOD: Respond to new agency authentication query	
1.3.7.1.1.1.1.2.1.1.1	OPERATOR(P): Hear the new agency's authentication query letters	
1.3.7.1.1.1.1.2.1.1.2	OPERATOR(C): Trace the query letters on the ACEOI	
1.3.7.1.1.1.1.2.1.1.3	OPERATOR(M): Respond to agency with correct letter	
1.3.7.1.1.1.1.2.1.2	METHOD: Query new agency for authentication	
1.3.7.1.1.1.1.2.1.2.1	OPERATOR(C): Choose new query letters on the ACEOI	
1.3.7.1.1.1.1.2.1.2.2	OPERATOR(M): Query the agency with the new letters	
1.3.7.1.1.1.1.2.1.2.3	OPERATOR(P): Hear the agency's response letter	
1.3.7.1.1.1.1.2.1.2.4	OPERATOR(C): Determine correctness of agency's response	
1.3.7.1.1.1.1.2.1.3	*SELECTION:* If agency responds incorrectly first time:	
1.3.7.1.1.1.1.2.1.3.1	METHOD: Use METHOD: Query new agency for authentication	
1.3.7.1.1.1.1.2.1.4	*SELECTION:* If agency responds incorrectly second time:	
1.3.7.1.1.1.1.2.1.4.1	OPERATOR(M): Attempt agency contact on secondary frequency	
1.3.7.1.1.1.1.2.1.4.2	METHOD: Use METHOD: Conduct DASC Check-in	
1.3.7.1.1.1.1.3	METHOD: Provide mission information	
1.3.7.1.1.1.1.3.1	METHOD: Report Friendly Situation	
1.3.7.1.1.1.1.3.2	METHOD: Brief activity using IFREP format	
1.3.7.1.1.1.1.3.2.1	OPERATOR(M): Pass Call sign	
1.3.7.1.1.1.1.3.2.2	OPERATOR(M): Pass Mission Number	

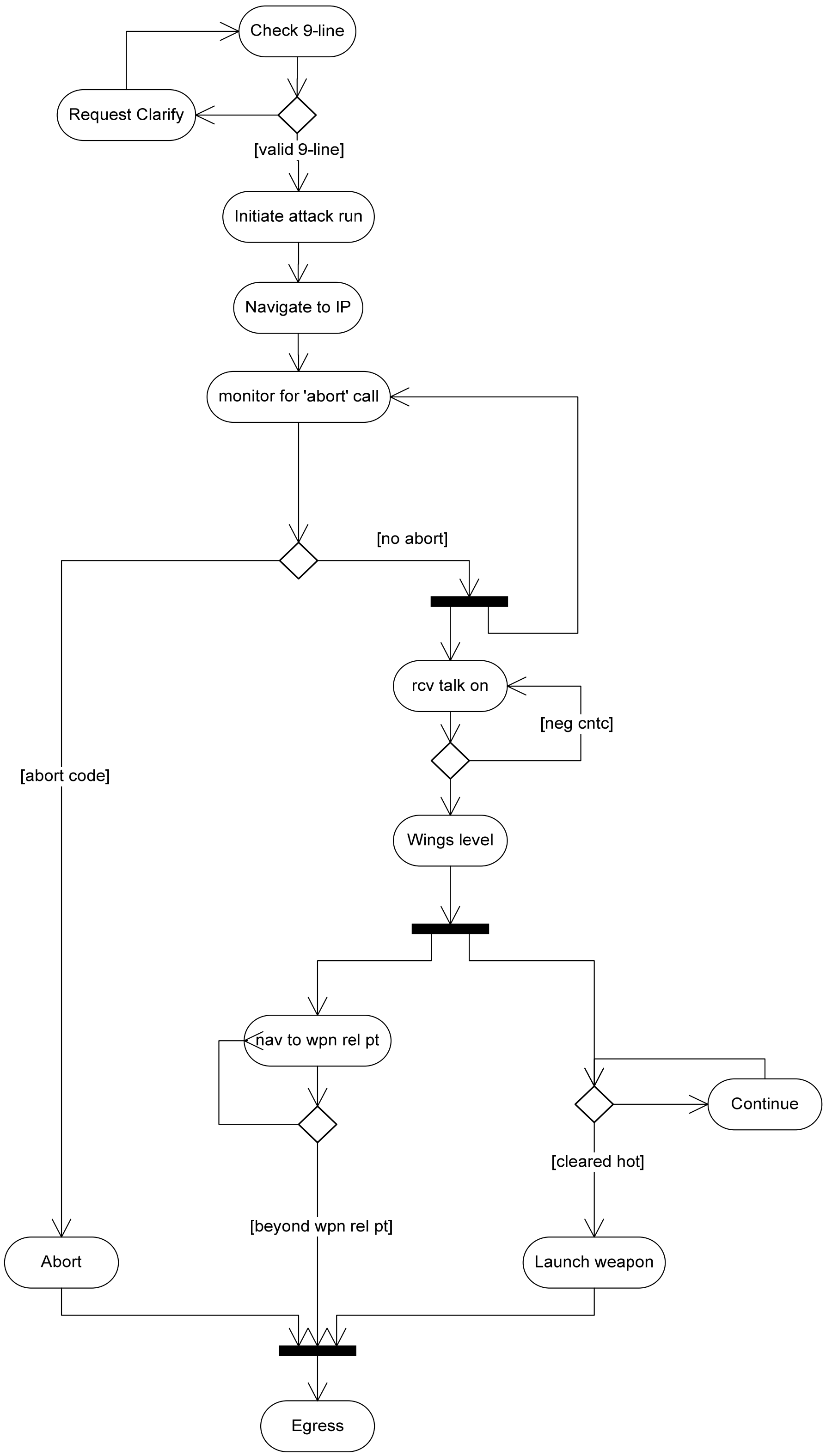
1.3.7.1.1.1.1.3.2.3	OPERATOR(M): Pass Request	
1.3.7.1.1.1.1.3.2.4	OPERATOR(M): Pass Location	
1.3.7.1.1.1.1.3.2.5	OPERATOR(M): Pass Time	
1.3.7.1.1.1.1.3.2.6	OPERATOR(M): Pass Results	
1.3.7.1.1.1.1.3.2.7	OPERATOR(M): Pass Remarks (Weather, BDA, Threat)	
1.3.7.1.1.1.1.3.3	OPERATOR(M): Pass Enemy Forces Remaining	
1.3.7.1.1.1.1.3.4	OPERATOR(M): Obtain Return Routing	
1.3.7.1.1.1.1.3.5	OPERATOR(M): Obtain Friendly and Enemy Situation Update for Route	
1.3.7.1.1.1.1.4	METHOD: Obtain friendly and enemy situation update for return route	
1.3.7.1.1.1.1.4.1	OPERATOR(P): Hear agency's situation report	
1.3.7.1.1.1.1.4.2	OPERATOR(M): Copy abbreviated report on kneeboard	
1.3.7.1.1.1.1.4.3	OPERATOR(C): Understand how information in report changes mission plan, if at all	
1.3.7.1.1.1.2	METHOD: Obtain Routing Information	
1.3.7.1.1.1.2.1	*SELECTION:* If agency provides routing information:	
1.3.7.1.1.1.2.1.1	METHOD: Receive routing information	
1.3.7.1.1.1.2.1.1.1	OPERATOR(P): Hear agency's routing instructions	
1.3.7.1.1.1.2.1.1.2	OPERATOR(M): Copy instructions on kneeboard	
1.3.7.1.1.1.2.1.1.3	OPERATOR(M): Denote pertinent information on map	
1.3.7.1.1.1.2.1.1.4	**GOAL:** Understand implications of flying the assigned route	
1.3.7.1.1.1.2.1.1.4.1	METHOD: Determine if assigned route crosses restrictive FSCMs or known enemy positions	
1.3.7.1.1.1.2.1.1.4.1.1	OPERATOR(P): Compare route CPs to map information	
1.3.7.1.1.1.2.1.1.4.2	METHOD: Determine if assigned routing causes unacceptable changes to mission plan	
1.3.7.1.1.1.2.1.1.4.2.1	OPERATOR(C): Calculate arrival time at terminal area	
1.3.7.1.1.1.2.1.1.4.3	*SELECTION:* If assigned route interferes with successful completion of mission	
1.3.7.1.1.1.2.1.1.4.3.1	METHOD: Obtain approval of modified route	
1.3.7.1.1.1.2.1.1.4.3.1.1	OPERATOR(C): Mentally formulate reason for route rejection	
1.3.7.1.1.1.2.1.1.4.3.1.2	OPERATOR(C): Choose alternate route from accumulated information and map CP data	
1.3.7.1.1.1.2.1.1.4.3.1.3	OPERATOR(M): Explain to DASC the need for route change and offer alternative route	
1.3.7.1.1.1.2.1.1.4.3.1.4	OPERATOR(P): Hear confirmation of approval for alternate route	

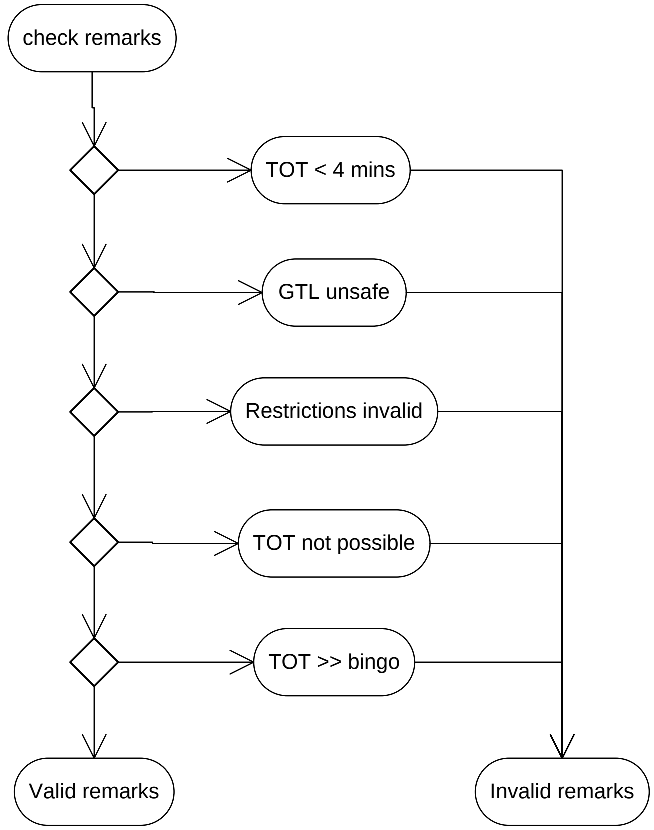
1.3.7.1.1.1.2.2	*SELECTION:* If agency does not provide routing information:	
1.3.7.1.1.1.2.2.1	OPERATOR(M): Request routing information	
1.3.7.1.1.1.2.2.2	METHOD: Use METHOD: Receive routing information	
1.3.7.1.1.2	*SELECTION:* If LOS does not exist with target agency:	
1.3.7.1.1.2.1	METHOD: Attempt alternate form of contact	
1.3.7.1.1.2.1.1	OPERATOR(M): Attempt contact with agency via other airborne assets	
1.3.7.1.1.2.1.2	OPERATOR(M): Attempt contact via ground nets	
1.3.7.1.1.2.1.3	OPERATOR(M): Increase altitude within tactical limits	
1.3.7.1.1.2.2	*SELECTION:* If contact is achieved via alternate methods:	
1.3.7.1.1.2.2.1	METHOD: Use METHOD: Conduct Liaison with DASC	
1.3.7.2	**GOAL:** Navigate to FOB or FARP	
1.3.7.2.1	OPERATOR(C): Visually match assigned route CPs to map CPs	
1.3.7.2.2	OPERATOR(C): Query copilot regarding understanding of the assigned route	
1.3.7.2.3	METHOD: Provide copilot with navigation data	
1.3.7.2.3.1	*SELECTION:* If time and workload permit:	
1.3.7.2.3.1.1	MAINTENANCE METHOD: Provide copilot with navigation updates	
1.3.7.2.3.1.1.1	*SELECTION:* If prominent terrain feature along route to next CP is visible	
1.3.7.2.3.1.1.1.1	OPERATOR(P): Identify prominent terrain feature	
1.3.7.2.3.1.1.1.2	OPERATOR(M): Advise copilot to fly toward prominent terrain feature	
1.3.7.2.3.1.1.2	*SELECTION:* If prominent terrain feature is not along route or not visible	
1.3.7.2.3.1.1.2.1	OPERATOR(C): Identify initial heading or cardinal direction to next CP	
1.3.7.2.3.1.1.2.2	OPERATOR(M): Advise copilot to fly a heading or cardinal direction	
1.3.7.2.3.2	*SELECTION:* If time and workload do not permit:	
1.3.7.2.3.2.1	METHOD: Provide copilot with digital navigation data	
1.3.7.2.3.2.1.1	OPERATOR(M): Enter route in navigation computer	
1.3.7.2.3.2.1.2	OPERATOR(M): Advise copilot that route is loaded in computer and task to fly route unassisted	
1.4	**GOAL:** Debrief Mission	X
1.4.1	**GOAL:** Report Observations	X
1.4.1.1	METHOD: Debrief with Unit Intelligence Section	
1.4.1.2	METHOD: Debrief with TACC or Group / Wing Operations Section	
1.4.2	**GOAL:** Evaluate Mission Plan	X
1.4.3	**GOAL:** Evaluate Mission Execution	X

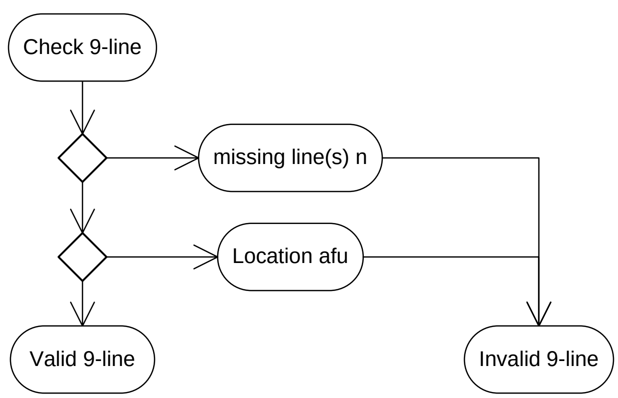
THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B. STATE TRANSITION DIAGRAMS




Check 9-line
Request Clarify
[valid 9-line]
Initiate attack run
Navigate to IP
monitor for 'abort' call
[no abort]
rcv talk on
[neg cntc]
[abort code]
Wings level
nav to wpn rel pt
Continue
[cleared hot]
[beyond wpn rel pt]
Abort
Launch weapon
Egress


check remarks
TOT < 4 mins
GTL unsafe
Restrictions invalid
TOT not possible
TOT >> bingo
Valid remarks
Invalid remarks


Check 9-line
missing line(s) n
Location afu
Valid 9-line
Invalid 9-line

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX C. ACRONYM AND TERM DEFINITIONS

Joint Terminal Attack Controller (JTAC). A JTAC is a qualified (certified) service member who, from a forward position, directs the action of combat aircraft engaged in CAS and other air operations.

Close Air Support. Air action by fixed and rotary wing aircraft against hostile targets that are in close proximity to friendly forces and that require detailed integration of each air mission with the fire and movement of those forces.

ACA. Airspace Coordination Area.

BDA. Battle Damage Assessment

BP. Battle Position (Rotary wing or GCE)

CAS. Close Air Support. Air action by three dimensional block of airspace in a target area, established by the appropriate ground commander, in which friendly aircraft are reasonably safe from friendly surface fires.

Air Officer (AO). At the battalion level, an officer who functions as the chief advisor to the battalion commander on all air operations matters. He also supervises the training and employment of the two battalion tactical air control parties (TACP). Also, ALO. Air Liaison Officer (USA)

AMC. Air Mission Commander

CBU. Cluster Bomb Unit

CFL. Coordinated Fire Line. A line beyond which conventional surface

fire support means (mortars, artillery, and naval surface fire support ships) may fire at any time within the boundaries of the establishing headquarters without additional coordination.

DASC. Direct Air Support Center (USMC). The principal air control agency of the US Marine Corps air command and control system responsible for the direction and control of air operations directly supporting the ground combat element. If conducted from an airborne platform, called DASC(A).

FAC. Forward Air Controller; Final Attack Cone

FDC. Fire Direction Center

FEBA. Forward Edge of the Battle Area.

FFA. Free Fire Area. A specific area into Folding-Fin Aerial Rocket

FFE. Fire For Effect

FIST. Fire Support Team. A team provided Forward Observer

FSC. Fire Support Coordinator

FSCC. Fire Support Coordination Center.

GBU Guided Bomb Unit

GP. General Purpose

GTL. Gun Target Line

HA. Holding Area

FLOT. Forward Line of Own Troops

FSCL. Fire Support Coordination Line. Facilitate the expeditious attack of surface targets of opportunity beyond the coordinating measure. The FSCL applies to all fires using any conventional ammunition. The inability to conduct coordination will not preclude an attack beyond the FSCL.

FSCM. Fire Support Coordinating Measure

HE. High Explosive / HEI. High Explosive Incendiary

ILLUM. Illuminating

IP. Initial Point

JTAC. Joint Terminal Attack Controller. A qualified (certified) Service member who, from a forward position, directs the action of combat aircraft engaged in close air support and other offensive operations.

LOAL. Hellfire Lock On After Launch

LOBL. Hellfire Lock On Before Launch

LST. Laser Spot Tracker

LTL. Laser to Target Line

MACCS. Marine Air Command and Control Military Grid Reference System

Mission Precedence. A designation System. assigned to a mission to indicate its priority or urgency of accomplishment. The following are the precedence listed in order of highest to lowest priority.

Emergency Mission. Mission involves safety of U.S. or other friendly lives or requires immediate transport of vital supplies or equipment or urgently required resupply ammunition or medical supplies.

Routine MEDEVAC. Evacuation of deceased personnel, a patient with a minor illness, or a patient requiring transfer between medical facilities for further treatment.

Mandatory Mission. Emergency in nature and involves possible loss of human life or national prestige to the extent that normally unacceptable risks will be taken in its accomplishment.

Priority Mission. Tactical movement of equipment or personnel whose excessive delay would jeopardize successful mission accomplishment. It includes logistic operations where delays would result in excessive material loss through spoilage or seizure by the enemy.

Routine Mission. Administrative or tactical transport of personnel or equipment, where time is not a critical factor and delay will not endanger lives or loss of material.

Urgent MEDEVAC. Evacuation of critically wounded, injured, or ill personnel who require early hospitalization and whose immediate evacuation is a matter of life or death.

Priority MEDEVAC. Evacuation of seriously wounded, injured, or ill personnel who require early hospitalization but whose immediate evacuation is not a matter of life or death.

NSFS. Naval Surface Fire Support

RFA. Restrictive Fire Area. An area in which specific restrictions are imposed and into which that exceed those restrictions will not be delivered without coordination with the establishing headquarters.

RFL. Restrictive Fire Line. Line established between converging friendly forces that prohibits fires, or effects from fires, across the line without coordination with the affected force.

MSL. Mean Sea Level

NFA. No Fire Area. An area designated by the appropriate commander into which fires or their effects are prohibited.

SDZ. Surface Danger Zone

SEAD. Suppression of Enemy Air Tactical Air Coordinator (Airborne)

TACC. Tactical Air Control Center (USN) Theater Air Control System

TAOC. Tactical Air Operations Center

TOT. Time On Target

TOW. Tube-launched, Optically tracked, Wire-Time To Target

WP. White Phosphorus

TAC(A). / Tactical Air Command Center (USMC). The principal Navy / USMC air command and control agency from which air operations and air defense warning functions are directed.

TACP. Tactical Air Control Party

APPENDIX D. GAME DESIGN DOCUMENT

A FAC(A) Battlefield Management Simulation

Game Design Document

VERSION NOTES

Items and text modified by a strike-through (e.g., ~~example strikethrough~~) have been designated as version 2 improvements and are included in this document only for completeness.

GAME OVERVIEW

Cleared Hot is a training application created to help solidify the understanding of how fire support platforms and their ordnance integrate in three dimensions. Cleared Hot is designed to support the mission qualification training of the Fleet squadron FAC(A) students.

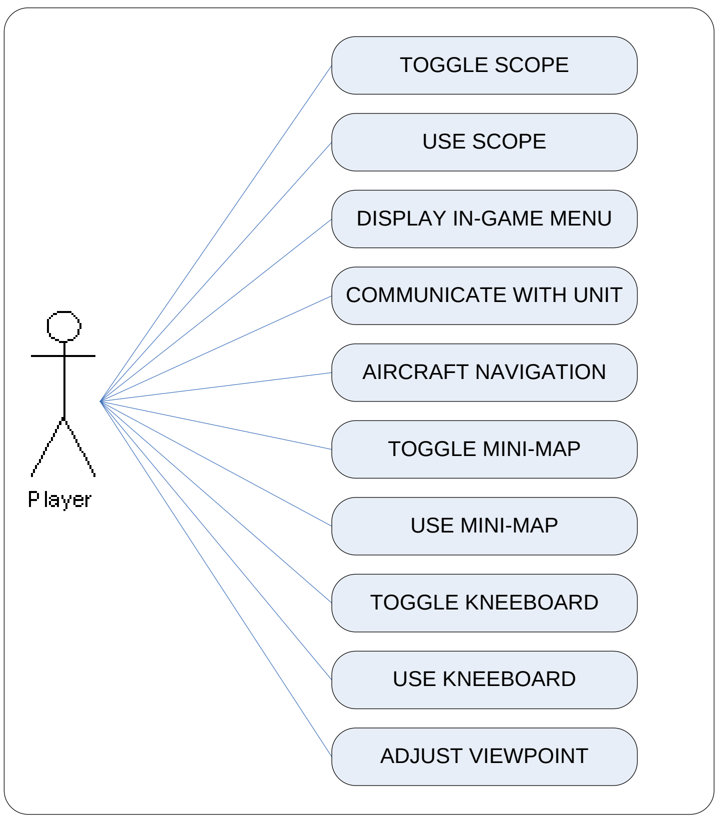
INTRODUCTION CINEMATIC

You begin the application and watch an introduction cinematic depicting an F-18 Hornet inbound from an Initial Point toward a column of enemy tanks in the desert, under the direction of a Forward Air Controller (Airborne). Scene changes briefly to show an enemy SA-6 nearby searching for the friendly jet. Scene changes back to the FAC(A) talking the eyes of the Hornet pilot onto the lead unit in the column, as radio traffic crackles with the message of a friendly artillery unit reporting “Shot, over.” The FAC(A) quickly acknowledges the artillery’s message that they have sent rounds downrange to suppress the SA-6. Quick pan to the jet as it rolls onto the final attack heading and reports “Wings level.” Shot of the FAC(A) searching the air for the jet, jet appears, and camera pans right and down to the tank column indicating the FAC(A) has determined that the jet has the correct target. New shot of the SA-6 as a full artillery sheath explodes around and on top of it. Voice over of the FAC(A) saying “Cleared Hot” as you view the F-18; a second later it releases a bomb. Pan to the tank column and watch the lead element explode. End cinematic.

MAIN GAME SCREEN

A.	Environment Panel
B.	Clock
C.	Out the Window
D.	Kneeboard Display Toggle
E.	Communication Unit Selector
F.	Communication Panel
G.	Mini-Map Display
H.	Equipment Selector
I.	Stack Diagram
J.	Radar
K.	Communication History
L.	Scope View

GAME INTERACTION SYSTEM



1. Toggle Scope
The Scope View is displayed by left clicking on the Scope View Toggle button.

2. Use Scope (see Scope System)

3. Display In-Game Menu
Display the in-game menu by pressing the Escape key.

4. Communicate with Unit (see Radio Dialog System)
In order to communicate with any entity, the user must click the communication button which is anchored to the left side of the out-the-window view. The result of this action brings the unit dialog widget into view (See Comm Widget Diagram). This widget has two areas: unit selection bar and dialog entry.

5. Aircraft Navigation (see Aircraft Navigation System)

6. Toggle Mini-Map
The user left clicks on the Display Mini-Map button. A second left click on the Display Mini-Map button will hide the Mini-Map.

7. Use Mini-map (see Mini-Map System)

8. Toggle Kneeboard
The user left clicks on the Display Kneeboard button. A second left click on the Display Kneeboard button will hide the Kneeboard.

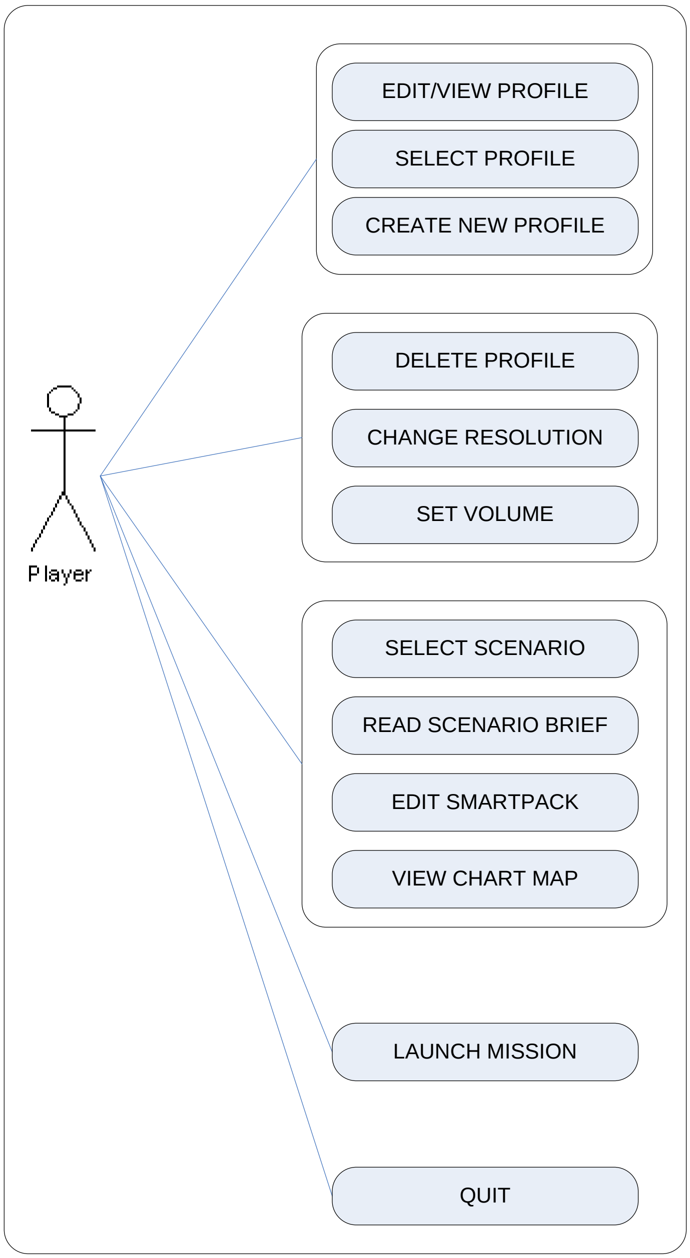
9. Use Kneeboard (see Kneeboard System)

10. Adjust Viewpoint
The user left clicks and drags in the Out the Window view. The view is rotated and pitched the same amount the cursor is dragged.

The view adjustments are limited to +/- 120 degrees horizontal and +40/-30 degrees vertical. If the horizontal limits are reached and the aircraft is in a hover, the aircraft will adjust its heading to match the amount the cursor dragged.

The current observer viewpoint is displayed in the Radar Display as a cone. The Viewpoint Adjustment can be snapped to center, by the User pressing the space bar.

MENU INTERACTION



1. Edit/View Profile
The user can change the Profile name and level of difficulty.

2. Select Profile
The user can select an existing Profile.

3. Create New Profile
The user can create a new profile by clicking on the Create New Profile button. The menu changes to the Edit/View Profile screen with default values already entered.

4. Delete Profile
The user can delete an existing Profile by (…)

5. Change Resolution
The user can change the resolution by selecting a screen resolution from the drop-down list and left clicking the Apply button. The application will immediately change the display and window resolution. (saved in the profile?)

6. Set Volume
The user can adjust the volume of the application by left click dragging the volume scrollbar.

7. Select Scenario
The user selects an existing scenario by left clicking on the scenario name in the scrollable list window.

8. Read Scenario Brief (SMEAC)
Separate buttons to view each piece of the SMEAC.

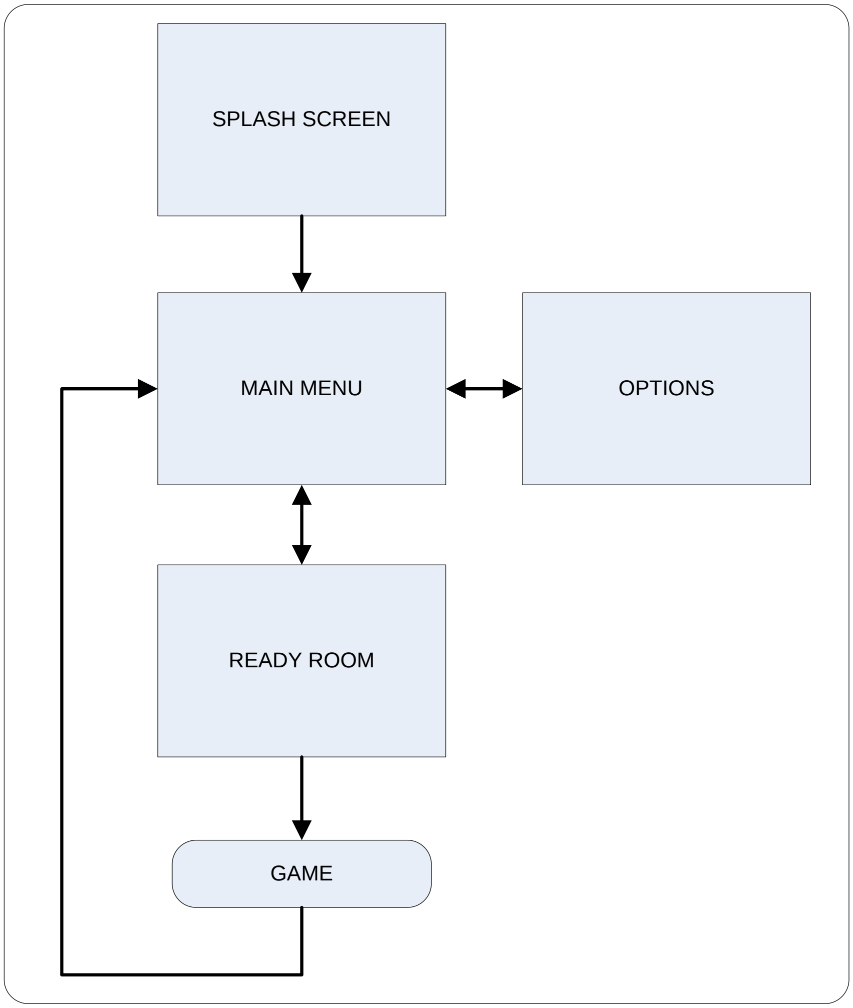
9. Edit Smart Pack

10. View Chart Map
By left clicking on the Chart Map button, the user can view the chart map that is preloaded with scenario specific information.

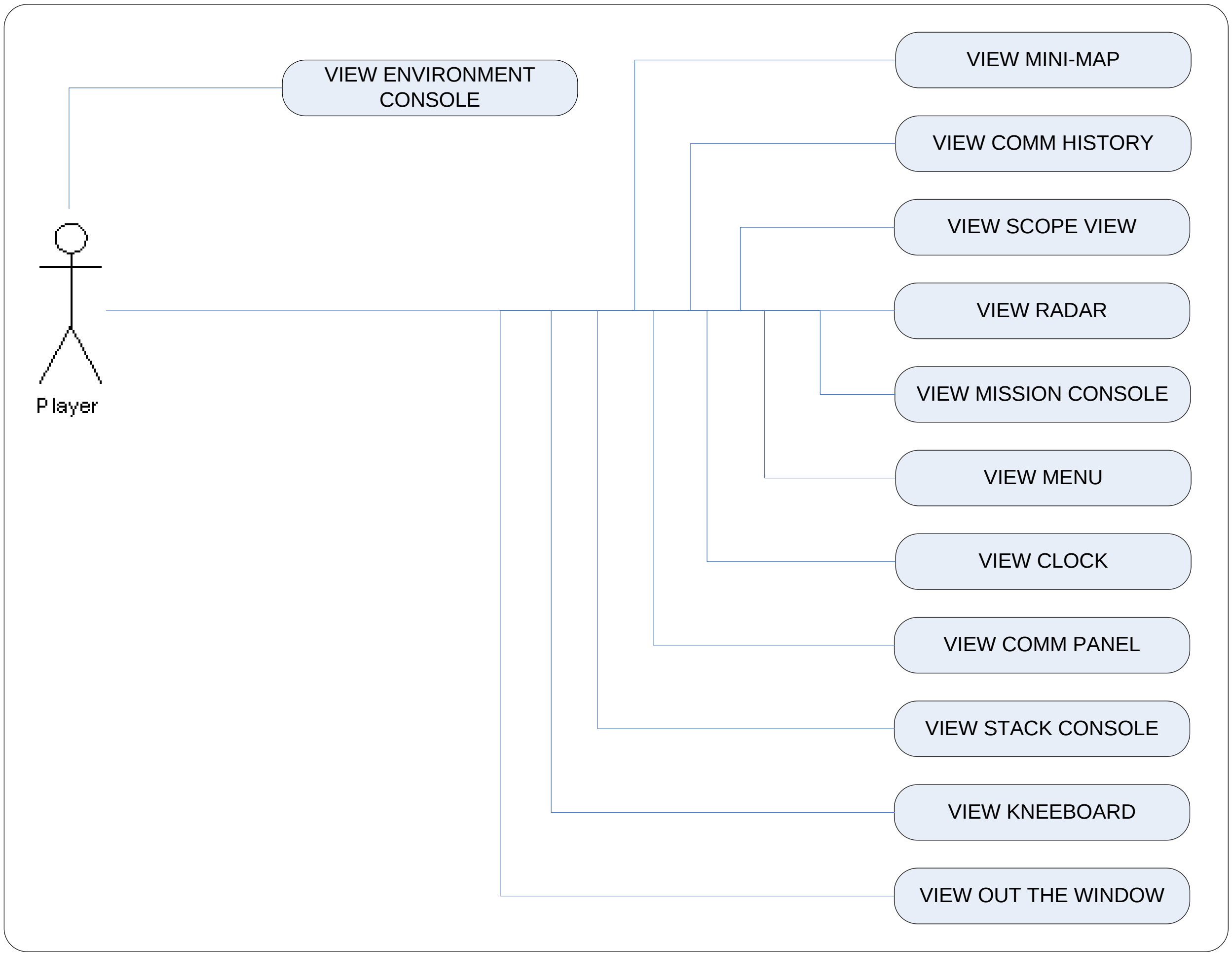
11. Launch Mission
The user can launch into the mission by left clicking on the Launch Mission button. This will cause the application to start loading the scenario.

12. Quit Application

MENU INTERACTION FLOW



DISPLAY SYSTEM



1. View Mini-Map (See Display Mini-Map System)

2. View Communication History
The Communication History is rendered as a scrollable list of text representing the last incoming and outgoing radio communications.

3. View Scope
The Scope view is rendered in place of the Out the Window pane and provides a zoomed-in view.

4. View Environment Console
The Environment Console displays the current weather conditions of the Scenario. The wind direction strength, cloud cover, and visibility are displayed here. The Environment data is displayed left and right of the Mission Clock.

5. View Radar

The Radar graphically depicts the area around the ownship from a top-down view. The user's aircraft icon is centered in the round panel with other units rendered in positions relative to the ownship. The outer ring of the Radar displays the compass rose with the ownship's heading at the 12 o'clock position. The edge of the outer ring represents a distance of 15 kilometers. Any units further away than 15km are "pinned" to the outer ring.

Friendly units are rendered blue, enemy units are rendered red, and unknown units are rendered grey. No icons are rendered until the user has performed a Check In with the Air Officer.

Units that are above the ownship's MSL are rendered as triangles, units below are rendered as an upside down triangle, and units within +150 /-150 ft are rendered as circles.

The user's current field of view direction is displayed as a "V" shape, oriented in the direction of view. There is no user interaction related to the Radar, it only displays information to the user.

6. View Mission Console

The Mission Console is rendered on the bottom ¼ of the screen. The Mission Console contains the Stack Console, the Radar, the Communication Unit Selector

7. View Shell Menu

8. View Mission Clock

Displays the local time of the environment.

9. View Communication Panel

10. View Stack Console
The Stack Console is automatically updated based on the status of the supporting aircraft. It displays an inverted triangle graphic with 4 horizontal lines. Each line can have text displayed on it.

There is a maximum of 3 different stack "pages", cycled by click on the next/previous buttons. Each stack page can list 4 different aircraft, one on each line.

11. View Kneeboard (See Display Kneeboard System)
The Kneeboard is a floating GUI rendered with multiple tabs.

12. View Out the Window
The Out the Window view depicts what the FAC(A) would see out the cockpit window (minus the cockpit itself). The view is a 3D perspective rendering containing the terrain, environment, and 3D models representing the units in the scenario.

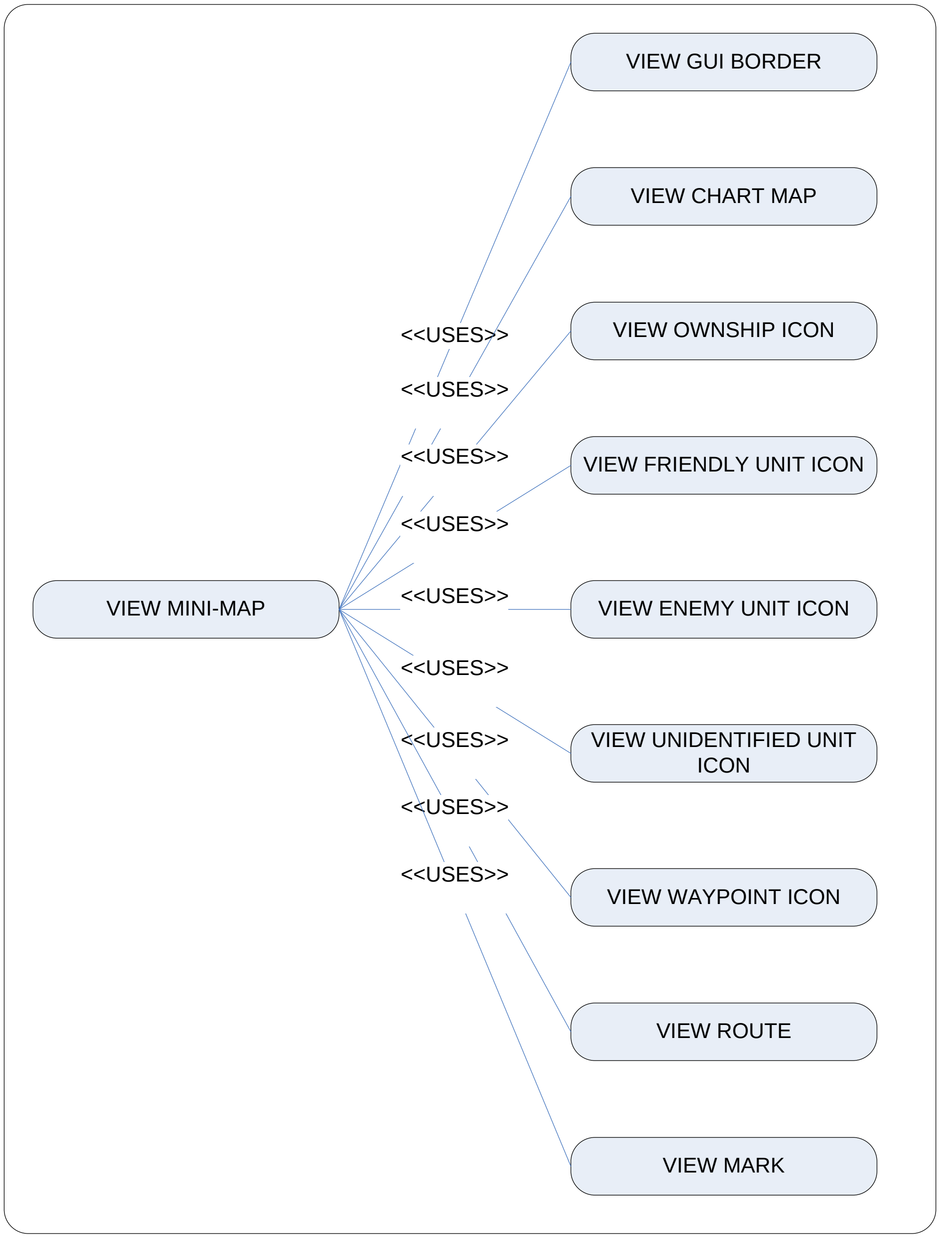
The Out the Window view should be rendered at 60 degrees field of view horizontal with a vertical field of view matched to keep the aspect ratio square with the application window.

The terrain is a pre-built terrain mesh utilizing satellite imagery as texture.

The unit objects are low resolution models. If the unit is currently selected, a colorful translucent sphere is rendered surrounding the unit.

Clouds, fog/haze, and time of day (dawn to dusk) are adjustable per scenario.

DISPLAY MINI-MAP SYSTEM



(See Mini-map Symbology for Icons)

1. View GUI Border
Render the window border, title bar, Zoom scrollbar, FollowOwnship button, Track up/North up toggle, and the Hover button.

2. View Chart Map
Render the chart map pertaining to the loaded scenario. The chart map is a 1:50k military chart covering the area of the scenario.

3. View Ownship Icon
Render the ownship icon as a helicopter symbol fixed in the center of the Mini-map. The ownship icon always renders.

4. View Friendly Icon
Friendly units appear on the Minimap only after the user Checks In with the Air Officer. The symbols are rendered blue with the correct MIL-SPEC symbol by default.

5. View Enemy Icon
Enemy units appear on the MiniMap only after the user Checks In with the Air Officer. The symbols are rendered red with the correct MIL-SPEC symbol by default.

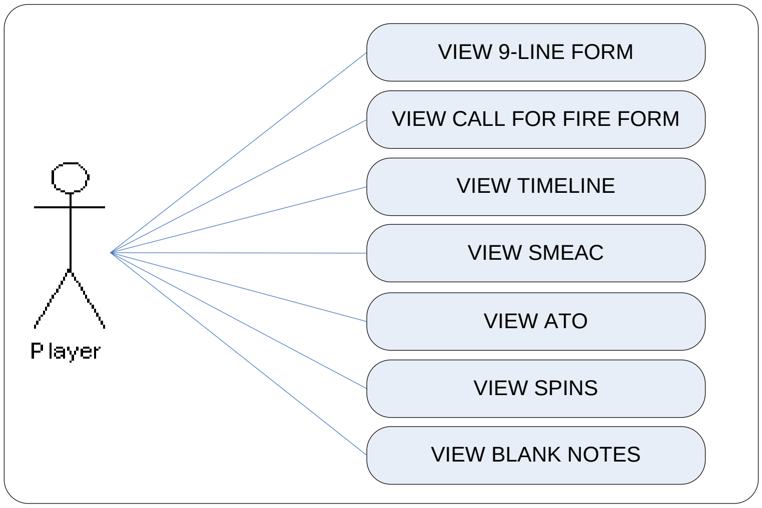
6. View Unidentified Icon
Render the unidentified unit icons in a grey color. No unidentified unit icons get displayed until the user Checks In with the Air Officer.

7. View Waypoint Icon
Render the waypoint icons (…)

8. View Route
Render the route as a straight line from one waypoint to the next. The route line should draw from the ownship symbol to the first symbol even if the ownhip is moving.

9. View Mark
Render a "mark" symbol on the chart map at the current location of the ownship.

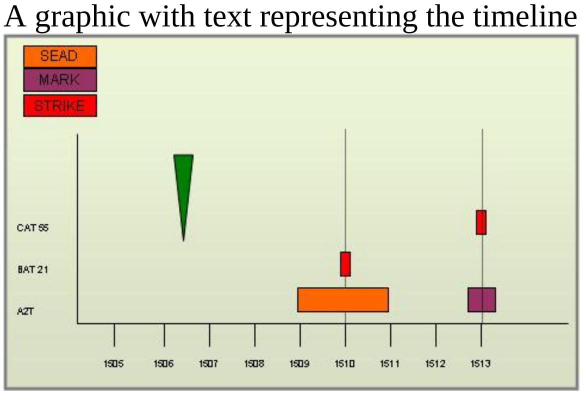
DISPLAY KNEEBOARD SYSTEM



1. View 9-line form
Contains 9 lines of information the user must supply via pull down and numeric entry fields.

2. View Call for Fire Form (see Call for Fire Options)
Contains 9 lines of information the user must supply via pull down and numeric entry fields.

3. View Timeline
A graphic with text representing the timeline of events in the mission.



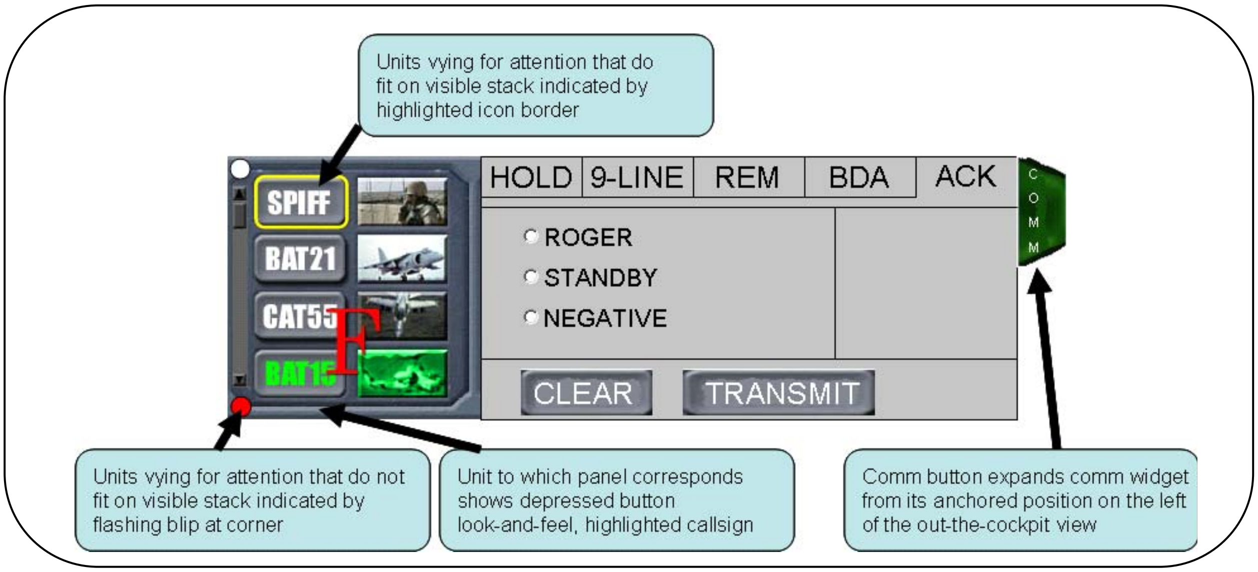
4. View SMEAC
Text display similar to the mission brief screen.

5. View ATO
Text display for Air Tasking Order; includes aircraft arrival, departure, ordnance, and mission.

6. View SPINS
Text display for SPecial INstructionS; includes locations of units.

7. View Blank Notes
Blank scrolling text box.

RADIO DIALOG INTERFACE



Comm Widget Diagram

Unit selection bar
A specific unit is selected for communication by pressing its respective pushbutton on the unit selection bar. There will always be a pushbutton depressed when utilizing the widget. Upon initial attempt at communication, the system defaults to the air officer pushbutton. Each pushbutton is overlaid with a graphic representative of the unit and displays the unit call sign. When a pushbutton is selected, its look and feel changes to a depressed button with backlit call sign. Only one pushbutton may be selected at a time; selecting an entity deselects the previously selected entity's pushbutton. The unit selection bar is populated from the ATO and SPINS; it contains a pushbutton for each CAS section, the Air Officer (AO), and the Fire Direction Center (FDC). There may be more units active during a game session than may be displayed at once on the unit selection bar. In this case, unit pushbuttons may be accessed by scrolling up or down. If a unit is scheduled to arrive on station but is not yet in the area, its pushbutton is grayed out. For example, if the current time is 1400 and Bat 21 is scheduled to arrive at 1415, its pushbutton is grayed out and non-selectable.

Two indicator lights on the top and bottom of the scroll bar provide a cue when a unit not currently visible on the widget is vying for attention. The top light blinks red when a 'needy' unit may be accessed by scrolling up; functionality is similar for the lower indicator light. When there are no 'needy' units, the indicator lights are white. If a unit needing attention is visible on the widget, its pushbutton is outlined with yellow highlight.

If the comm widget is in the retracted position (i.e., nothing is visible except for the communication button) and a unit is vying for attention, then the "COMM" label on the expansion tab blinks red.

Dialogue Entry Window

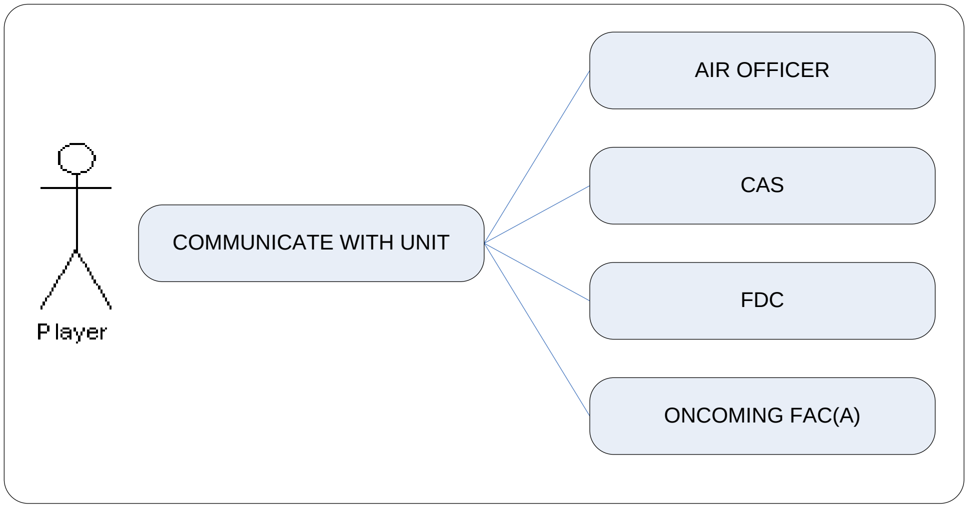
The area to the right of the unit selection bar displays communication options; options are context-sensitive. For example, the options available for messages to pass to a CAS unit are different from those available to pass to the Air Officer. Message types are organized into type groups; major types are accessible by tabs as shown in the Comm Widget Diagram above. Within each major type, the user may set individual parameters, and parameters are either mandatory or optional. For example, a major type of message for a CAS unit is "HOLD." When preparing a hold instruction, the user must select a holding stack point and altitude. These are mandatory items. Additionally, the user may want to advise the CAS unit of other friendly aircraft in the area. Thus, there is also an optional parameter section in which the user may select from existing friendly units.

Global buttons are present on the bottom of the dialogue entry window. These buttons are labeled "CLEAR" and "TRANSMIT" and have the functionality of Clearing all selections made for the current window, and of transmitting the Message, respectively. The global transmit button is grayed out until all mandatory parameters are selected

If the user makes any changes to a dialogue entry window, those changes are persistent, even if the user switches to another unit's dialogue entry window or transmits the message to the current unit. For example, a user selects a holding checkpoint and altitude for Bat 21, then before transmitting that message, switches over to Bat 24 and sends him a message. When the user comes back to Bat 21, the previously selected holding point and altitude are still displayed on the dialogue entry window.

RADIO DIALOG SYSTEM

Communicate With Unit Use Cases



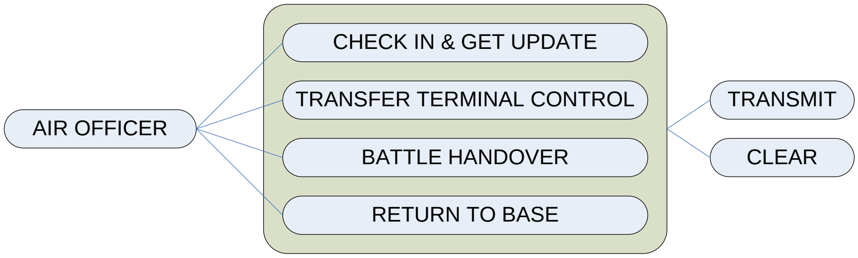
Four agencies are available for communication; they are the Air Officer, the CAS, the FDC, and the Oncoming FAC(A). This section describes their dialogue branches. While there may be multiple CAS units active during a game session, all of their dialogue branches are identical, and so are generalized here for brevity. There is only one Air Officer, FDC, and Oncoming FAC(A) unit in a game session.

In the following pages, the dialogue branches are organized to ‘peel the onion’ on each agency’s dialogue choices. The first part of an agency’s section will show the major message types available to a player, accompanied by a brief description of the purpose of each type.

Subsequent sections for each agency show further branches for each major message type, indicating which values may be sent within a message. Following this graph of the message branches, there is a template that shows the message structure, along with an example of that type of message. Templates and examples for game agent responses are included.

All variables used in the template may be referenced in the section ‘Radio Dialog Variables’ toward the end of this document.

AIR OFFICER MAJOR MESSAGE TYPES



Check in / Get Update
The user selects this option to check in with the Air Officer and request an update to the current situation.

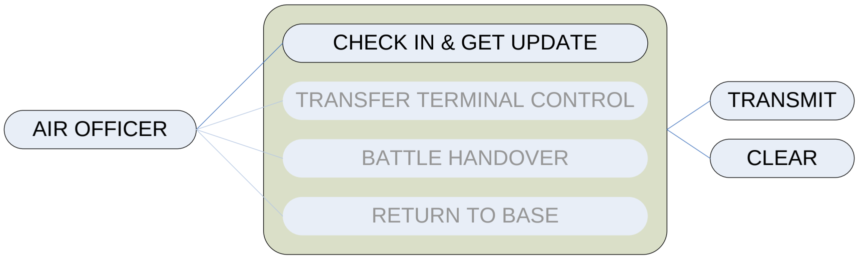
Request Terminal Control
The user selects this option to inform the Air Officer that they are prepared to take terminal control of the objective area.

Submit Attack Package
The user selects this option to send the pre-defined Attack plan to the Air Officer. If the Attach Package is valid, the Air Officer will reply with a “roger”.

Conduct Battle Handover
The user selects this option to pass a situation brief that includes all changes to friendly and enemy unit positions and attacks conducted since the user took terminal control. It is meant to increase situational awareness in the next terminal controller.

Return To Base
The user selects this option to inform the Air Officer they are returning to base. This option effectively ends the current mission.

Air Officer Message Branch: Check In & Get Update



Template

Player: “<CALLSIGN_AIR_OFFICER>, this is <CALLSIGN_PLAYER>, mission <MISSION_NUM_PLAYER>, up as fragged at <CP_INITIAL_PLAYER>, cherubs 2 with universal. Playtime (45 – <MINUTES_SINCE_LAUNCH>). Request friendly and enemy situation update.”

Air Officer: “Roger <CALLSIGN_PLAYER>, copy up as fragged. Push to <CP_RECON_PLAYER> and stand by for situation update.”
if(there exists at least one friendly artillery unit){
“Be advised”
}
for(i=0; i<number of artillery units; i++){
“<CALLSIGN_ARTY_FRIENDLY(i)> located at
<GRID_ARTY_FRIENDLY(i)>, gun target line
<GUNLINE_ARTY_FRIENDLY(i)>.”
}

(TIME DELAY 10 SECONDS)

“<CALLSIGN_PLAYER>, <CALLSIGN_AIR_OFFICER>,
<SITUATION_AIR_OFFICER>”

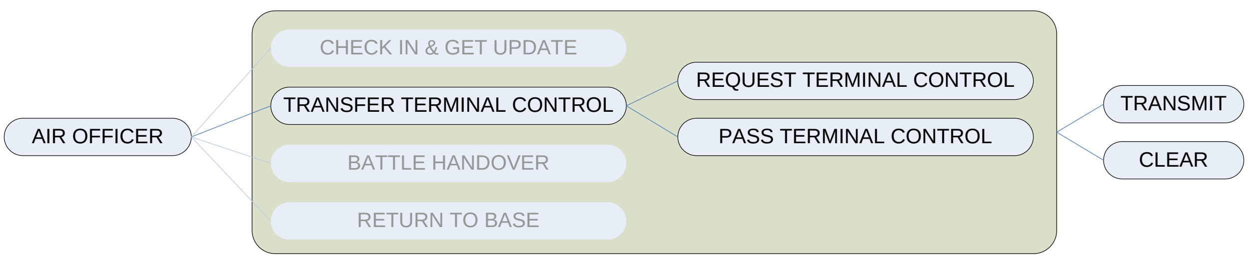
Example

Player: “Mongo, this is Viper 22, mission number 3014, up as fragged at Bush, cherubs 2 with universal. Playtime 0 + 45. Request friendly and enemy situation update.”

Air Officer: “Roger Viper 22, copy up as fragged. Push to Star and stand by for situation update. Be advised R7M located at NU 682 187, gun target line 065.”

(TIME DELAY: 10 SECONDS)	
Air Officer:	“Viper 22, Mongo, 1st Battalion, 7th Marines was ambushed by a mechanized infantry battalion while in pursuit of the enemy after they retreated from their assault through Noble Pass. Bravo company is encountering heavy resistance in vicinity of grid NU 740 230. They're engaged with the enemy’s lead elements in the open. The CO wants to break the enemy lines so Bravo can push through to envelop the ambush forces. Viking 20 is a section of Hornets out of Mina Al Palms; they should be arriving on station any minute for CAS. R7M is in direct support of 1/7 with six guns. My position is 7 clicks southwest of Spider, and I do not have eyes on. I need continuous coverage over the engagement area. You will need to run Viking out of the south. Scouts have sighted 2 x ZSU-23-4 seven clicks northeast of Cloud in the wash behind the enemy front. Map datum is WGS-84; all players on universal. Type one CAS in effect. My FAC, callsign Beaker, is moving up to Bravo’s position and may be ready to direct fires at the end of your playtime. He'll be up TAD-5 and local; I will be monitoring.”

Air Officer Message Branch: Transfer Terminal Control



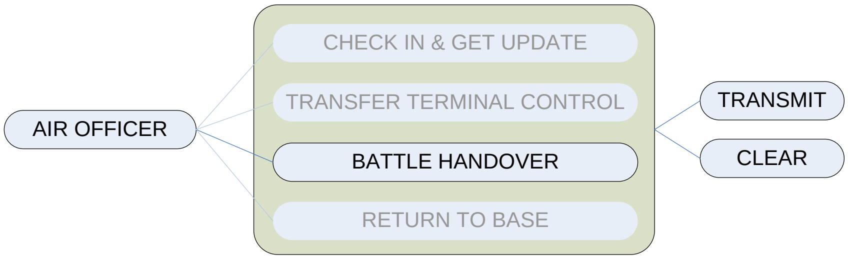
Template

Player:	“<CALLSIGN_AIR_OFFICER>, this is <CALLSIGN_PLAYER>, ready to accept terminal control.”
Air Officer:	“<CALLSIGN_PLAYER>, <CALLSIGN_AIR_OFFICER> you have terminal control.”

Example

Player:	“Mongo, this is Viper 22, ready to accept terminal control.”
Air Officer:	“Viper 22, Mongo, you have terminal control.”

Air Officer Message Branch: Conduct Battle Handover



Template

Player: "<CALLSIGN_AIR_OFFICER>, <CALLSIGN_PLAYER>, battle handover brief to follow. Enemy situation:"

if(numReportsADA > 0){

for(int i = 0; i < numReportsADA; i++){

"<FOCUS_EN_ADA_DESC(i)> at <FOCUS_EN_ADA_LOC(i)>, last active at <FOCUS_EN_ADA_ACTIVE_TIME(i)>

}

}else{

"No significant enemy ADA detected."

}

if(numReportsEnAcft > 0){

for(int i = 0; i < numReportsEnAcft; i++){

"<FOCUS_EN_ACFT_DESC(i)> at <FOCUS_EN_ACFT_LOC(i)>, seen at <FOCUS_EN_ACFT_TIME_SIGHTED(i)>."

}

else{

"No significant air threat detected."

}

if(numReportsEnGrndUnit > 0){

for(int i = 0; i < numReportsEnGrndUnit; i++){

"<FOCUS_EN_GRND_UNIT_DESC(i)> at <FOCUS_EN_GRND_UNIT _LOC(i)>, sighted at <FOCUS_EN_GRND_UNIT_TIME_SIGHTED(i)>."

}

}else{

"No enemy ground units in the immediate area."

}

if(<COMPILED_BDA> != null){

"BDA to follow: <COMPILED_BDA>."

}

"Friendly situation:"

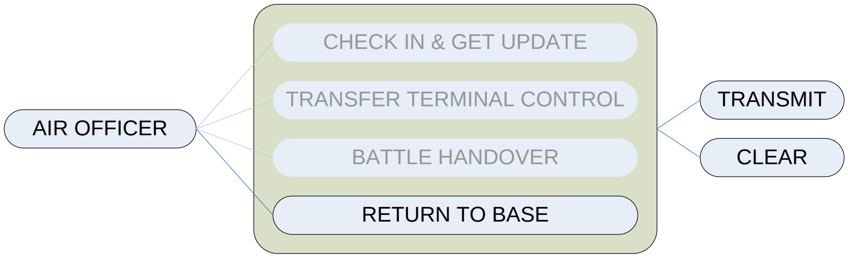
```
                if(numReportsFrGrndUnit > 0){
                        for(int i = 0; i < numReportsFrGrndUnit; i++){
                                "<FOCUS_FR_GRND_UNIT_TITLE(i)> is at
                                <FOCUS_FR_GRND_UNIT_LOC(i)>
                                if(FOCUS_FR_GRND_UNIT_TYPE == ARTY){
                                        "with GTL <FOCUS_GRD_UNIT_GTL(i)>."
                                }
                        }
                }else{
                        "No friendly ground forces in the terminal area."
                }
                "Mission:"
                if(num9Lines > 0){
                        for(int i = 0; i < num9Lines; i++){
                                "<FOCUS_9_LINE_CAS_CALLSIGN(i)> is set up to run
                                out of <FOCUS_9_LINE_IP(i)> on
                                <FOCUS_9_LINE_TARGET_DESC(i) in vicinity of
                                <FOCUS_9_LINE_GRID(i)> at TOT
                                <FOCUS_9_LINE_TOT(i)>."
                        }
                }
                if(numCFF > 0){
                        for(int i = 0; i < numCFF; i++){
                                "<FOCUS_CFF_UNIT_CALLSIGN(i)> is conducting a
                                <FOCUS_CFF_MISSION_TYPE(i)> mission targeting
                                <FOCUS_CFF_TARGET_DESC(i)> in vicinity of
                                <FOCUS_CFF_GRID(i)>"
                                if(<FOCUS_CFF_MISSION_TYPE> == SEAD){
                                        "in support of  <FOCUS_CFF_CAS_CS >."
                                }
                        }
                }
                if(numCAS > 0){
                        "On station is"
                        for(int i = 0; i < numCAS; i++){
                                "<CALLSIGN_CAS(i)>, mission <MISSION_CAS(i)>, a
                                <CAS_UNIT_SIZE_COMMON_NAME(i)> of
                                <CAS_ACFT_TYPE_COMMON(i)> stacked at
                                <STACK_CP_CAS(i)> angels
                                <ALTITUDE_CAS(i)>/1000, playtime
                                (<BINGO_CAS(i)> - <SYSTEM_TIME>)."
                        }

Air Officer:    "<CALLSIGN_PLAYER>, <CALLSIGN_AIR_OFFICER>, copy all."
```

Example

Player:	“Mongo, Viper 22, battle handover brief to follow. Enemy situation: ZSU-23-4 at NU 720 265, last active at 1512. No significant air threat detected. Chipotlean mechanized battalion at NU 725 245, sighted at 1500. BDA to follow: 7 T-72 destroyed at NU 735 650 at 1345. Friendly situation: Bravo 1/7 is at NU 700 890. R7M is at NU 750 450 with GTL 065. Mission: Wake 30 is set up to run out of Chevy on T-72 MBT in vicinity of NU 653 253 at TOT 1454. R7M is conducting a SEAD mission targeting ZSU-23-4 in vicinity of NU 745 689 in support of Wake 30. On station is Bat 10, mission 3014, a section of Hornets stacked at Chevy angels 20, playtime 0+20, and Viking 20, mission 3016, a section of Hornets stacked at Chevy angels 22, playtime 0+30.”
Air Officer:	“Viper 22, Mongo, copy all.”

Air Officer Message Branch: Return To Base



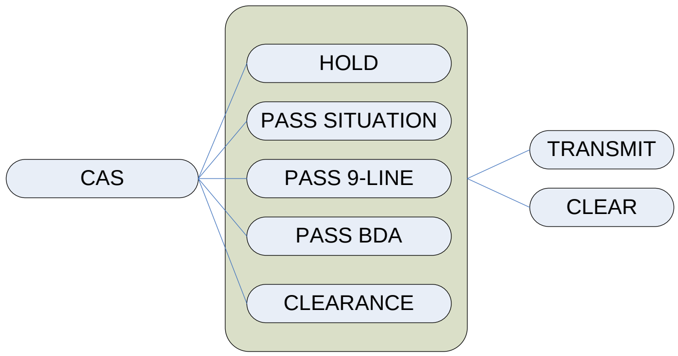
Template

Player:	“<CALLSIGN_AIR_OFFICER>, <CALLSIGN_PLAYER>, RTB.”
Air Officer:	“<CALLSIGN_PLAYER>, <CALLSIGN_AIR_OFFICER>, roger.”

Example

Player:	“Mongo, Viper 22, RTB.”
Air Officer:	“Viper 22, Mongo, roger.”

CAS MAJOR MESSAGE TYPES



Hold
The user selects this option in order to pass holding instructions to the CAS.

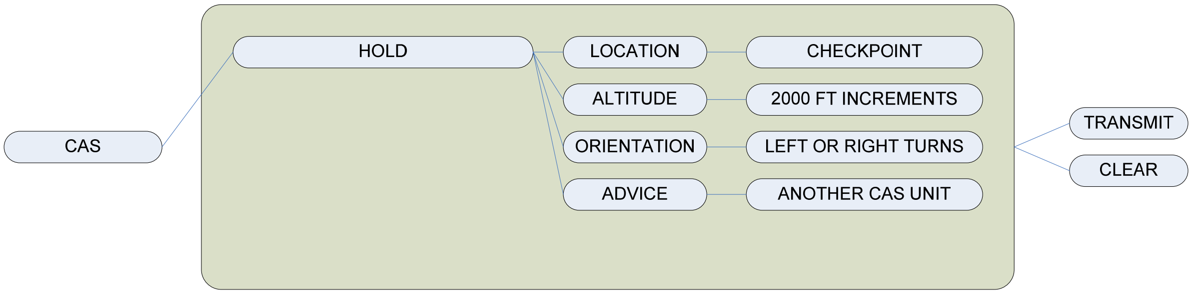
Pass Situation
The user selects this option to build the situational awareness of the CAS regarding the current friendly and enemy situation, the airborne assets currently on station, and the general flow of assets in the target area.

Pass 9-Line
The user selects this option in order to pass an Air Officer approved 9-line (which includes remarks) to the CAS for execution.

Pass BDA
The user selects this option in order to notify the CAS of the effectiveness of the attack just executed. BDA is given for the flight, not for individual aircraft in the flight.

Pass Clearance
The user selects this option in order to clear or abort CAS.

CAS Message Branch: Hold



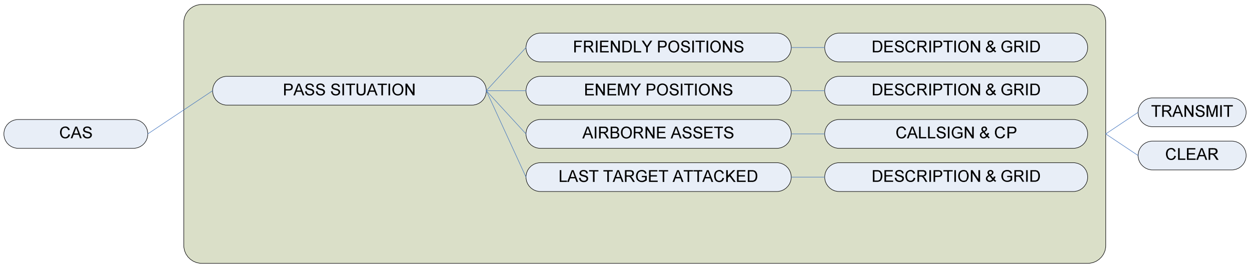
Template

Player:	“<CALLSIGN_CAS>, this is <CALLSIGN_PLAYER>, hold at <CAS_ASSIGNED_STACK_CP>, angels (<CAS_ALT> / 1000), <ORIENT> turns,” be advised <ADVISE>. Report established. Stand by for situation update.”
CAS:	“Viper 22, Bat 10, wilco.”

Example

Player:	“Bat 10, this is Viper 22, hold at Star, angels 19, left hand turns. Be advised Viking 20 anchored at Star, angels 17. Report established. Stand by for situation update.”
CAS:	“Viper 22, Bat 10, wilco.”

CAS Message Branch: Pass Situation



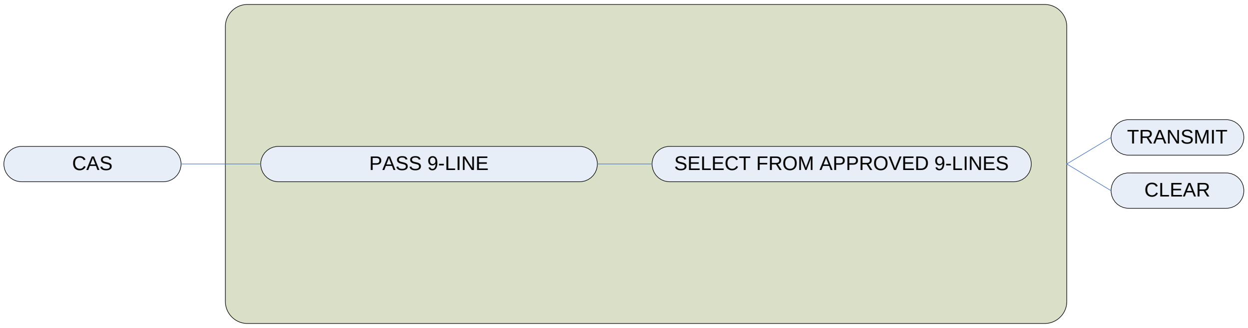
Template

Player:	“<CALLSIGN_CAS>, this is <CALLSIGN_PLAYER>, <SELECTED_FRIENDLY_UNIT_NAME_ABBREV(i)> is located at <SELECTED_FRIENDLY_UNIT_GRID(i)>. <SELECTED_ENEMY_UNIT_TYPE(i)> is vicinity of <SELECTED_ENEMY_UNIT_GRID(i)>. CAS on station is <FRIENDLY_AIR_UNIT_CALLSIGN(i)>, a <FRIENDLY_AIR_UNIT_QTY_DESC(i)> of <FRIENDLY_AIR_UNIT_TYPE_COMMON(i)> anchored at <FRIENDLY_AIR_UNIT_ASSIGNED_CP(i)>. Last target attacked was a <LAST_ENEMY_UNIT_ATTACKED_DESC> at <LAST_ENEMY_UNIT_ATTACKED_GRID>. Stand by for 9-line.”
CAS:	“Viper 22, Bat 10, copy all.”

Example

Player:	“Bat 10, this is Viper 22, Bravo company is located at NU 740 230. R7M is located at NU 682 187. ZSU-23-4 is vicinity of NU 700 255, ZSU-23-4 is vicinity of NU 720 265. CAS on station is Viking 20, a section of Hornets anchored at Frog, and Wake 30, a section of Harriers anchored at Charger. Last target attacked was a BMP-2 at NU 723 246. Stand by for 9-line”
CAS:	“Viper 22, Bat 10, copy all.”

CAS Message Branch: Pass 9-Line



Note: Approved ninelines are those that have been filtered through the Air Officer. Only ninelines screened in this manner will be available for selection on this tab.

Template

Player: "<CALLSIGN_CAS>, <PLAYER_CALLSIGN>, nine line to follow: <FOCUS_NINELINE_IP>, <FOCUS_NINELINE_HDG>, <FOCUS_NINELINE_DIST>."

(TIME DELAY 3 SECONDS)

"<FOCUS_NINELINE_ELEV>, <FOCUS_NINELINE_TGT_DESC>, <FOCUS_NINELINE_TGT_GRID>."

(TIME DELAY 3 SECONDS)

"<FOCUS_NINELINE_MARK_TYPE>, <FOCUS_NINELINE_FRIENDLY_LOC>, egress <FOCUS_NINELINE_EGRESS>."

(TIME DELAY 3 SECONDS)

"<FOCUS_NINELINE_SEAD_DESC>. Final attack cone <FOCUS_NINELINE_CONE>. <FOCUS_NINELINE_RESTRICTIONS>."

CAS: "<CALLSIGN_CAS> copies <FOCUS_NINELINE_ELEV>, <FOCUS_NINELINE_GRID>. Final attack cone <FOCUS_NINELINE_CONE>. <FOCUS_NINELINE_RESTRICTIONS>."

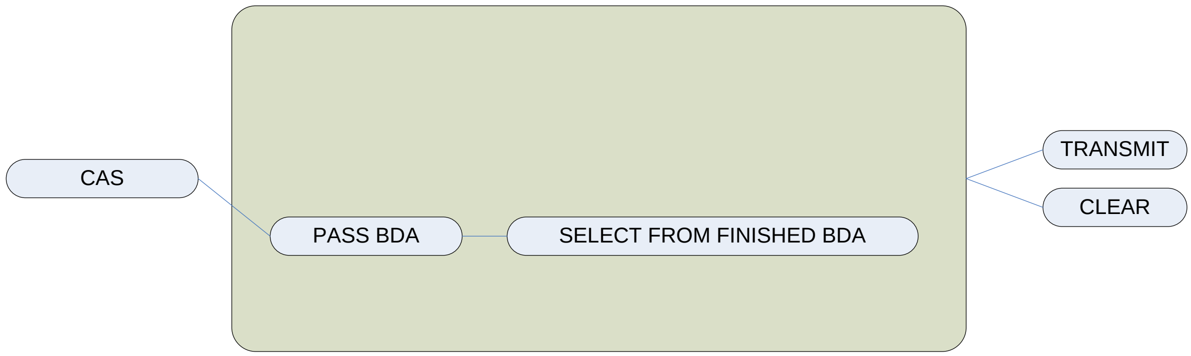
Player: "<CALLSIGN_CAS>, <PLAYER_CALLSIGN>, TOT <FOCUS_NINELINE_TOT>."

CAS:	“<CALLSIGN_CAS> copies TOT <FOCUS_NINELINE_TOT>.”

Example

Player:	“Bat 10, Viper 22, nine line to follow. Gambler, 320, 12.5.”
(TIME DELAY 3 SECONDS)	
	“400, BMP-2 in the open, NU 723 246.”
(TIME DELAY 3 SECONDS)	
	“Willy Pete, southeast 2000, egress east to Charger, then southwest to Frog.”
(TIME DELAY 3 SECONDS)	
	“Continuous suppression of ZSU-23-4 at NU 700 255 from 53 to 55, gun target line 005. Final attack cone 330° to 360°. Remain east of 72 easting.”
CAS:	“Bat 10 copies 400, NU 723 246. Final attack cone 330° to 360°. Remain east of 72 easting.”
Player:	“Bat 10, Viper 22, TOT 54.”
CAS:	“Bat 10 copies TOT 54.”

CAS Message Branch: Pass BDA



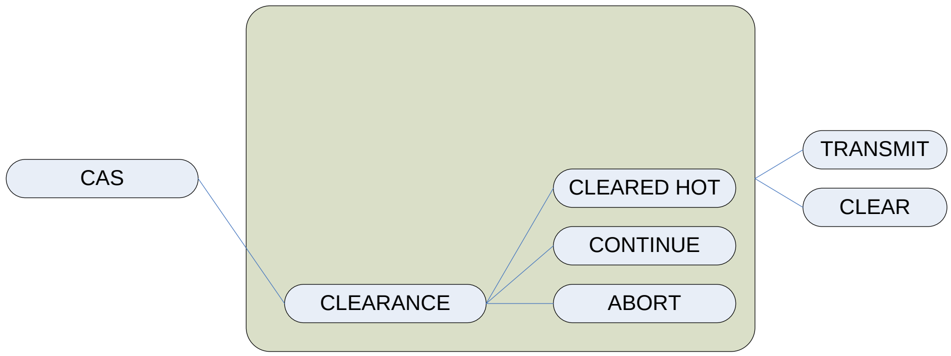
Template

Player:	“<CALLSIGN_CAS>, <CALLSIGN_PLAYER>, BDA to follow. <FOCUS_BDA_NUM_TARGET> <FOCUS_BDA_TYPE_TARGET> <FOCUS_BDA_DAMAGE_TARGET> at <FOCUS_BDA_GRID_TARGET> at <FOCUS_BDA_TIME_IN>.”
CAS:	“<CALLSIGN_PLAYER>, <CALLSIGN_CAS>, copy BDA.”

Example

Player:	“Wake 30, Viper 22, BDA to follow. 2 T-72 destroyed at NU 723 246 at 1454.”
CAS:	“Viper 22, Wake 30, copy BDA.”

CAS Message Branch: Pass Clearance



Template

Player:	“<CALLSIGN_CAS>, <CALLSIGN_PLAYER>,” *switch(SELECTION){* *case CLEARED_HOT:* “cleared hot.” *case CONTINUE:* “continue.” *case ABORT:* “abort.” *}*
CAS:	“<CALLSIGN_CAS>,” *switch(SELECTION){* *case CLEARED_HOT:* “in hot.” *case CONTINUE:* “continue.” *case ABORT:* “abort.” *}*

Example 1: CLEARED HOT

Player:	“Wake 30, Viper 22, cleared hot.”
CAS:	“Wake 30 in hot.”

Example 2: CONTINUE

Player:	“Wake 30, Viper 22, continue.”
CAS:	“Wake 30, continue.”

Example 3: ABORT

Player:	“Wake 30, Viper 22. Abort.”

CAS:	“Wake 30 abort.”

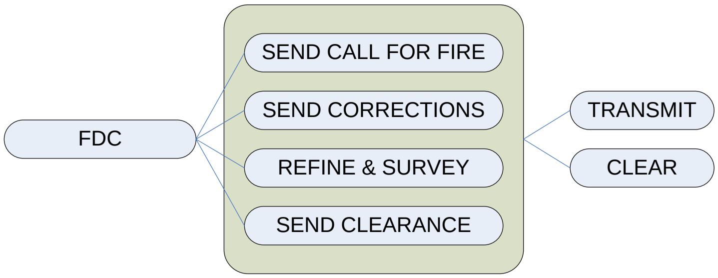
Some quick notes to be fleshed out later…

When a unit conducts an unsolicited call (not in response to some user comm), then clicking that unit’s pushbutton displays the ‘Acknowledge’ tab on the unit dialogue display. If the user selects ‘Roger’ and clicks ‘Transmit,’ then the unit never calls the user with that particular prompt again. If the user ignores the unit call, then the unit will wait for 30 seconds in most cases, and then send the call again.

We also need a hotkey assigned to automate the ‘Roger’ call, such a the keyboard letter ‘R.’ Hitting ‘R’ after a CAS unit made an unsolicited call has the effect of selecting the unit pushbutton, clicking ‘Roger’ and clicking ‘Transmit.’

The exception to waiting 30 seconds is wings level calls. When a unit calls ‘Wings level’ and the user ignores the call (does not select a clearance type), then the unit will wait only 5 seconds before prompting the user again. The unit will prompt every 5 seconds until it has reached the point where it could not successfully drop ordnance. At that point, the CAS unit prompts the user with “<FAC(A)_NAME>, <CAS_UNIT_NAME> egressing <FIRST_EGRESS_POINT>, no drop.”

FDC MAJOR MESSAGE TYPES



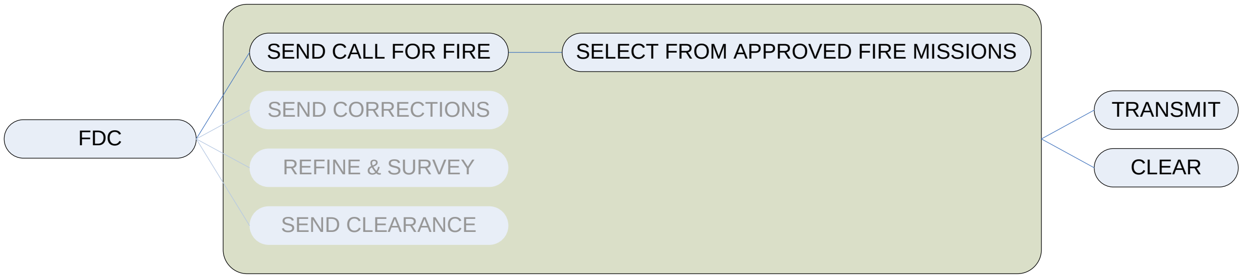
Call For Fire
The user selects this option in order to pass a fire mission to an artillery or mortar unit via the FDC (Fire Direction Center). A fire mission details certain target specifications much like a 9-line.

Corrections
The user selects this option to tell the FDC how far from the target the last round impacted.

Refinement & Surveillance
The user selects this option to complete the fire mission. The last rounds impacted near enough the target such that only minor corrections are required. The user tells the firing unit how accurate and sufficient were the rounds, and specifies if he wants the impact point recorded for future use.

Send Clearance
The user selects this path to either abort a mission in progress (before the first round is fired), or to give the command to fire or fire for effect. The fire command may not be used until the firing unit has reported “Ready.” Alternatively, the fire for effect command may not be used until after the first round of that mission has impacted the terrain.

FDC Message Branch: Send Call For Fire



Template 1: SEAD

Player: "<CALLSIGN_FDC>, <CALLSIGN_PLAYER>, <FIRE_MISSION_TYPE>, over."

****revisit this to incorporate FDC readbacks for all calls****

(TIME DELAY 3 SECONDS)

"Grid to suppress <GRID_TARGET>, grid to mark <GRID_TARGET_MARK>, over."

(TIME DELAY 3 SECONDS)

"<TARGET_DESCRIPTION> <TARGET_ACTIVITY>, <FIRE_METHOD_ENGAGE>, <FIRE_METHOD_CONTROL>, over."

(TIME DELAY 30 SECONDS)

FDC: "Message to observer, SEAD, <CALLSIGN_FDC>, grid to mark, illum on the deck, target number <TARGET_MARK_REG_NUM>, grid to suppress, target number <TARGET_SUPPRESS_REG_NUM>, time of flight <ARTY_ROUND_TOF_SECONDS> seconds, over."

Example 1: SEAD

Player: "R7M, this is Viper 22, SEAD, over.

(TIME DELAY 3 SECONDS)

"Grid to suppress NU 700 255, grid to mark NU 723 246, over."

(TIME DELAY 3 SECONDS)

"ZSU-23-4 in the open, continuous, TOT 54, over."

(TIME DELAY 30 SECONDS)

FDC: “Message to observer, SEAD, R7M, grid to mark, illum on the deck, target number AA1002, grid to suppress, target number AA1003, time of flight 30 seconds, over.”

Template 2: ADJUST FIRE

Player: “<CALLSIGN_FDC>, <CALLSIGN_PLAYER>, <FIRE_MISSION_TYPE>, over.”

(TIME DELAY 3 SECONDS)

“Grid <GRID_TARGET>, over.”

(TIME DELAY 3 SECONDS)

“<TARGET_DESCRIPTION> <TARGET_ACTIVITY>, <FIRE_METHOD_ENGAGE>, <FIRE_METHOD_CONTROL>, over.”

(TIME DELAY 30 SECONDS)

FDC: “<CALLSIGN_PLAYER>, <CALLSIGN_FDC>, message to observer, over.”

(TIME DELAY 3 SECONDS)

“<CALLSIGN_FDC>, 1 round, <TARGET_REGISTRATION_NUM>, <ARTY_ROUND_TOF_SECONDS> seconds, over.”

(TIME DELAY 30 SECONDS)

FDC: “<CALLSIGN_PLAYER>, <CALLSIGN_FDC>, <FIRE_MISSION_UNIT> is ready.”

Example 2: ADJUST FIRE

Player: “R7M, this is Viper 22, Adjust Fire, over.

(TIME DELAY 3 SECONDS)

“Grid NU 723 246, over.”

(TIME DELAY 3 SECONDS)

“ZSU-23-4 in the open, HE/Quick, at my command, over.”

(TIME DELAY 30 SECONDS)

FDC: “Viper 22, this is R7M, message to observer, over.”

(TIME DELAY 3 SECONDS)

“R7M, 1 round, AA1002, 30 seconds, over.”

(TIME DELAY 30 SECONDS)

FDC: “Viper 22, R7M, battery is ready.”

Template 3: FIRE FOR EFFECT

Player: “<CALLSIGN_FDC>, <CALLSIGN_PLAYER>, <FIRE_MISSION_TYPE>, over.”

(TIME DELAY 3 SECONDS)

“Grid <GRID_TARGET>, over.”

(TIME DELAY 3 SECONDS)

“<TARGET_DESCRIPTION> <TARGET_ACTIVITY>, <FIRE_METHOD_ENGAGE>, over.”

(TIME DELAY 30 SECONDS)

FDC: “<CALLSIGN_PLAYER>, <CALLSIGN_FDC>, message to observer, over.”

(TIME DELAY 3 SECONDS)

“<CALLSIGN_FDC>, 1 round, <TARGET_REGISTRATION_NUM>, <ARTY_ROUND_TOF_SECONDS> seconds, over.”

Example 3: FIRE FOR EFFECT

Player: “R7M, this is Viper 22, Fire For Effect, over.

(TIME DELAY 3 SECONDS)

“Grid NU 723 246, over.”

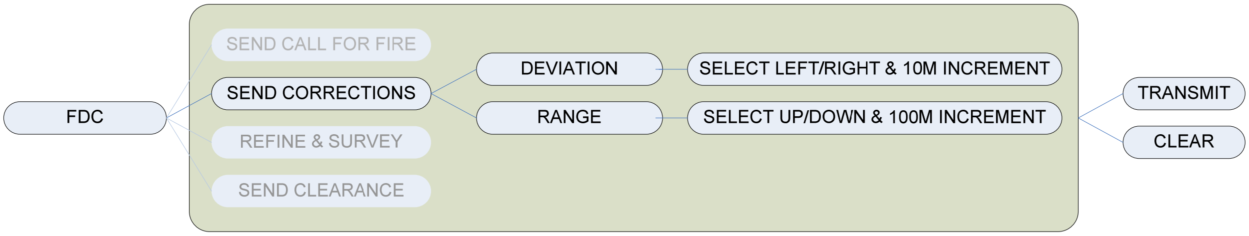
(TIME DELAY 3 SECONDS)

“ZSU-23-4 in the open, HE/Quick, over.”

(TIME DELAY 30 SECONDS)

FDC:	"Viper 22, this is R7M, message to observer, over."
(TIME DELAY 3 SECONDS)	
	"R7M, 1 round, AA1002, 30 seconds, over."

FDC Message Branch: Send Corrections



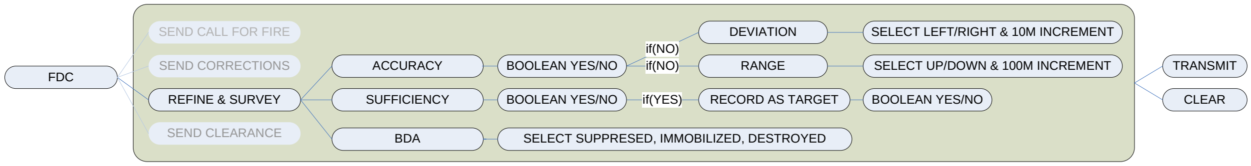
Template

Player:	"<CALLSIGN_FDC>, <CALLSIGN_PLAYER>," *if(deviation != 0){* "<FIRE_MISSION_CORRECTION_DEVIATION_TYPE> <FIRE_MISSION_DEVIATION_AMT>" *}* *if(range != 0){* "<FIRE_MISSION_CORRECTION_RANGE_TYPE> <FIRE_MISSION_CORRECTION_RANGE_AMT>" *}* "over."
FDC:	"<CALLSIGN_PLAYER>, <CALLSIGN_FDC>" *if(deviation != 0){* "<FIRE_MISSION_CORRECTION_DEVIATION_TYPE> <FIRE_MISSION_DEVIATION_AMT>" *}* *if(range != 0){* "<FIRE_MISSION_CORRECTION_RANGE_TYPE> <FIRE_MISSION_CORRECTION_RANGE_AMT>" *}* "out."

Example

Player:	"R7M, this is Viper 22, left 200, add 100, over."
FDC:	"Viper 22, R7M, left 200, add 100, out."

FDC Message Branch: Refinement & Surveillance



Template 1: ACCURATE & SUFFICIENT

Player: “<CALLSIGN_FDC>, <CALLSIGN_PLAYER>,”
if(record_as_target){
“record as target,”
}
“end of mission, target <TARGET_BDA>, over.”

FDC: readback***

Example 1: ACCURATE & SUFFICIENT

Player: “R7M, this is Viper 22, end of mission, target suppressed, over.”

FDC: readback***

Template 2: INACCURATE & SUFFICIENT

Player: “<CALLSIGN_FDC>, <CALLSIGN_PLAYER>”
if(deviation != 0){
“<FIRE_MISSION_CORRECTION_DEVIATION_TYPE>
<FIRE_MISSION_DEVIATION_AMT>”
}
if(range != 0){
“<FIRE_MISSION_CORRECTION_RANGE_TYPE>
<FIRE_MISSION_CORRECTION_RANGE_AMT>”
}
if(record_as_target){
“record as target,”
}
“end of mission, target <TARGET_BDA>, over.”

FDC: readback***

Example 2: INACCURATE & SUFFICIENT

Player: “R7M, this is Viper 22, right 30, add 50, end of mission, target suppressed, over.”

FDC: readback***

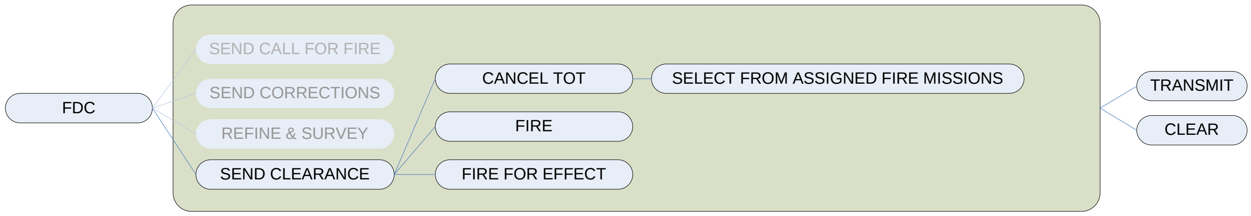
Template 3: INSUFFICIENT

Player:	"<CALLSIGN_FDC>, <CALLSIGN_PLAYER>" *if(inaccurate){* *if(deviation != 0){* "<FIRE_MISSION_CORRECTION_DEVIATION_TYPE> <FIRE_MISSION_DEVIATION_AMT>" *}* *if(range != 0){* "<FIRE_MISSION_CORRECTION_RANGE_TYPE> <FIRE_MISSION_CORRECTION_RANGE_AMT>" *}* *}* "repeat, over."
FDC:	readback***

Example 3: INSUFFICIENT

Player:	"R7M, this is Viper 22, right 30, add 50, repeat, over."
FDC:	readback***

FDC Message Branch: Send Clearance



Template 1: CANCEL TOT

Player:	"<CALLSIGN_FDC>, <CALLSIGN_PLAYER>, cancel TOT <SEAD_TOT> for SEAD mission, over."
FDC:	readback***

Example 1: CANCEL TOT

Player:	"R7M, this is Viper 22, cancel TOT 54 for SEAD mission, over."
FDC:	readback***

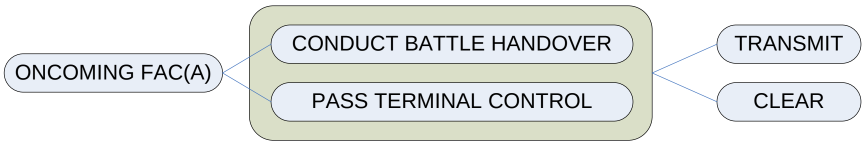
Template 2: FIRE || FIRE FOR EFFECT

Player:	"<CALLSIGN_FDC>, <CALLSIGN_PLAYER>" *if(fire){* "fire" *}else*{ "fire for effect" *}* "over."
FDC:	readback***

Example 2: FIRE || FIRE FOR EFFECT

Player:	"R7M, this is Viper 22, fire for effect, over."
FDC:	readback***

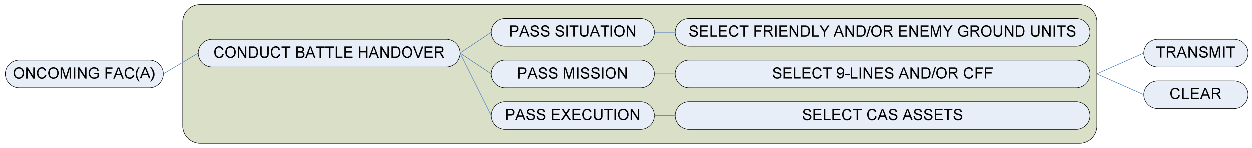
ONCOMING FAC(A) MAJOR MESSAGE TYPES



Conduct Battle Handover

Pass Terminal Control

Oncoming FAC(A) Message Branch: Conduct Battle Handover



Template

Player: "<CALLSIGN_AIR_OFFICER>, <CALLSIGN_PLAYER>, battle handover brief to follow. Enemy situation:"

if(numReportsADA > 0){

for(int i = 0; i < numReportsADA; i++){

"<FOCUS_EN_ADA_DESC(i)> at <FOCUS_EN_ADA_LOC(i)>, last active at <FOCUS_EN_ADA_ACTIVE_TIME(i)>

}

}else{

"No significant enemy ADA detected."

}

if(numReportsEnAcft > 0){

for(int i = 0; i < numReportsEnAcft; i++){

"<FOCUS_EN_ACFT_DESC(i)> at <FOCUS_EN_ACFT_LOC(i)>, seen at <FOCUS_EN_ACFT_TIME_SIGHTED(i)>."

}

else{

"No significant air threat detected."

}

if(numReportsEnGrndUnit > 0){

for(int i = 0; i < numReportsEnGrndUnit; i++){

"<FOCUS_EN_GRND_UNIT_DESC(i)> at <FOCUS_EN_GRND_UNIT _LOC(i)>, sighted at <FOCUS_EN_GRND_UNIT_TIME_SIGHTED(i)>."

}

}else{

"No enemy ground units in the immediate area."

}

if(<COMPILED_BDA> != null){

"BDA to follow: <COMPILED_BDA>."

}

"Friendly situation:"

if(numReportsFrGrndUnit > 0){

for(int i = 0; i < numReportsFrGrndUnit; i++){

"<FOCUS_FR_GRND_UNIT_TITLE(i)> is at <FOCUS_FR_GRND_UNIT_LOC(i)>

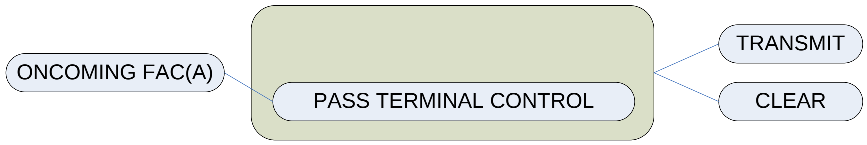
```
                    if(FOCUS_FR_GRND_UNIT_TYPE == ARTY){
                        "with GTL <FOCUS_GRD_UNIT_GTL(i)>."
                    }
                }
        }else{
                "No friendly ground forces in the terminal area."
        }
        "Mission:"
        if(num9Lines > 0){
                for(int i = 0; i < num9Lines; i++){
                    "<FOCUS_9_LINE_CAS_CALLSIGN(i)> is set up to run
                    out of <FOCUS_9_LINE_IP(i)> on
                    <FOCUS_9_LINE_TARGET_DESC(i) in vicinity of
                    <FOCUS_9_LINE_GRID(i)> at TOT
                    <FOCUS_9_LINE_TOT(i)>."
                }
        }
        if(numCFF > 0){
                for(int i = 0; i < numCFF; i++){
                    "<FOCUS_CFF_UNIT_CALLSIGN(i)> is conducting a
                    <FOCUS_CFF_MISSION_TYPE(i)> mission targeting
                    <FOCUS_CFF_TARGET_DESC(i)> in vicinity of
                    <FOCUS_CFF_GRID(i)>"
                    if(<FOCUS_CFF_MISSION_TYPE> == SEAD){
                        "in support of  <FOCUS_CFF_CAS_CS >."
                    }
                }
        }
        if(numCAS > 0){
                "On station is"
                for(int i = 0; i < numCAS; i++){
                    "<CALLSIGN_CAS(i)>, mission <MISSION_CAS(i)>, a
                    <CAS_UNIT_SIZE_COMMON_NAME(i)> of
                    <CAS_ACFT_TYPE_COMMON(i)> stacked at
                    <STACK_CP_CAS(i)> angels
                    <ALTITUDE_CAS(i)>/1000, playtime
                    (<BINGO_CAS(i)> - <SYSTEM_TIME>)."
                }

Oncoming FAC(A):   "<CALLSIGN_PLAYER>,
<CALLSIGN_ONCOMING_FAC(A)>, copy all."
```

Example

Player:	"Viper 18, Viper 22, battle handover brief to follow. Enemy situation: ZSU-23-4 at NU 720 265, last active at 1512. No significant air threat detected. Chipotlean mechanized battalion at NU 725 245, sighted at 1500. BDA to follow: 7 T-72 destroyed at NU 735 650 at 1345. Friendly situation: Bravo 1/7 is at NU 700 890. R7M is at NU 750 450 with GTL 065. Mission: Wake 30 is set up to run out of Chevy on T-72 MBT in vicinity of NU 653 253 at TOT 1454. R7M is conducting a SEAD mission targeting ZSU-23-4 in vicinity of NU 745 689 in support of Wake 30. On station is Bat 10, mission 3014, a section of Hornets stacked at Chevy angels 20, playtime 0+20, and Viking 20, mission 3016, a section of Hornets stacked at Chevy angels 22, playtime 0+30."
Oncoming FAC(A):	"Viper 22, Viper 18, copy all."

Oncoming FAC(A) Branch: Pass Terminal Control



Template

Player:	“<CALLSIGN_ONCOMING_FAC(A)>, this is <CALLSIGN_PLAYER>, you have terminal control.”
Oncoming FAC(A):	“<CALLSIGN_PLAYER>, <CALLSIGN_ONCOMING_FAC(A)> has terminal control.”

Example

Player:	“Viper 28, this is Viper 22, you have terminal control.”
Oncoming FAC(A):	“Viper 22, Viper 28 has terminal control.”

SYSTEM GENERATED MESSAGES

This section lists all the messages sent to the text display from agents during the game. These messages are unsolicited, i.e., they are event driven. Organization is by the game agent that sends the message. Each section includes a synopsis of the message content, the trigger, a template, and an example message. In some cases the message will be generated in response to another one sent by a different game agent, and if so that will be noted by the trigger.

AIR OFFICER MESSAGES

Response to arriving CAS when player has not taken terminal control

This is a short conversation that occurs between the Air Officer and a CAS unit that has just appeared at its spawn point. Its purpose is to remind the player that CAS is ready for work and that he needs to be assuming terminal control as soon as possible.

Trigger

Message from CAS - Check in when player has not taken terminal control

Template

Air Officer:	“<CALLSIGN_CAS>, <CALLSIGN_AIR_OFFICER>, roger. Copy <CAS_MISSION_CAPABILITY>. Stack at <CAS_SPAWN_CP> angels <CAS_ASSIGNED_ALT> and stand by. <PLAYER_CALLSIGN> will be assuming terminal control shortly.”

Example

Air Officer:	“Bat 10, Mongo, roger. Copy up as fragged. Stack at Bush angels 16 and stand by. Viper 22 will be assuming terminal control shortly.”

Prompt to player to accept terminal control

This is a reminder from the Air Officer to the player that CAS is ready for work and that he needs to be assuming terminal control as soon as possible.

Trigger

At least one CAS unit is on station, player has not taken terminal control, and 5 minutes have elapsed since either CAS spawned or since last message of this type.

Template

Air Officer:	“<PLAYER_CALLSIGN>, <CALLSIGN_AIR_OFFICER>. <CALLSIGN_CAS> is on station at <CAS_SPAWN_CP>. Understand you are ready to assume terminal control?”

Example

Air Officer:	“Viper 22, Mongo. Bat 10 is on station at Bush. Understand you are ready to assume terminal control?”

Passing approval or disapproval for an attack package

The Air Officer is assumed to be ‘listening’ to all conversations taking place on the Tactical Air Direction (TAD) net. Given this assumption, it is believable that the Air Officer will initiate unsolicited messages to the player indicating whether a plan the player had passed previously is approved or disapproved for execution.

Trigger

Attack plan passed to a firing unit meeting certain logic filters

Template

Air Officer:	“<CALLSIGN_PLAYER>, <CALLSIGN_AIR_OFFICER>, that plan for <PACKAGE_UNIT_FIRING(i)>” *for(i = 1; i < units firing; i++){* “and <PACKAGE_UNIT_FIRING(i)>” *}* “is <PACKAGE_APPROVAL_STATUS>.” *if(<PACKAGE_APPROVAL_STATUS> == TRUE){* “Your mark, your control.” *}else{* “<PACKAGE_FAILURE_REASON>.” *}*

Example 1: APPROVAL

Air Officer:	“Viper 22, Mongo, that plan for Wake 30 and R7M is approved. Your mark, your control.”

Example 2: DISAPPROVAL

Air Officer:	“Viper 22, Mongo, that plan for Wake 30 and R7M is not approved. Friendly units are inside danger close.”

Aborting an attack package

The Air Officer will abort any firing units that have not already been aborted by the Player if only 2 minutes remains until TOT.

Trigger

Attack plan has been disapproved by Air Officer && Player has not aborted a firing unit && System time + 2 minutes = TOT.

Template

Air Officer:	*for(i = 0; i < units firing; i++){* *if(<PACKAGE_UNIT_FIRING(i) == FDC){* "<CALLSIGN_FDC>, <CALLSIGN_AIR_OFFICER>, cancel TOT <SEAD_TOT> for SEAD mission, over." *}else{* "<CALLSIGN_CAS>, <CALLSIGN_AIR_OFFICER>, abort, over." *}*

Example

Air Officer:	"R7M, Mongo, cancel TOT 54 for SEAD mission, over. Wake 30, Mongo, abort, over."

CAS MESSAGES

Check in when player has not taken terminal control

This is a sequence wherein the newly-spawned CAS contacts the Air Officer to check in.

Trigger

CAS spawns and player has not taken terminal control.

Template

CAS:	"<CALLSIGN_AIR_OFFICER>, <CALLSIGN_CAS> is mission number <CAS_MISSION_NUM>, <CAS_MISSION_CAPABILITY> at <CAS_SPAWN_CP>, angels <CAS_ALT> / 1000, playtime <CAS_TOS>."

Example

CAS:	"Mongo, Bat 10 is mission number 3004, up as fragged at Bush, angels 20, playtime 0 + 30."

Check in when player has taken terminal control

This is a sequence wherein the newly-spawned CAS contacts the player to check in.

Trigger

CAS spawns and player has taken terminal control.

Template

CAS:	"<PLAYER_CALLSIGN>, <CALLSIGN_CAS> is mission number <CAS_MISSION_NUM>, <CAS_MISSION_CAPABILITY> at <CAS_SPAWN_CP>, angels <CAS_ALT> / 1000, playtime <CAS_TOS>."

Example

CAS:	"Viper 22, Bat 10 is mission number 3004, up as fragged at Bush, angels 20, playtime 0 + 30."

Arrival at stack point

This is a confirmation from the CAS to the player that CAS has arrived at his assigned stack point.

Trigger

CAS arrives at assigned stack point and altitude, as calculated by the game engine based on the time the player passed the command to that CAS to stack there.

Template

CAS:	"<PLAYER_CALLSIGN>, <CALLSIGN_CAS> established at <CAS_ASSIGNED_STACK_CP>, angels <CAS_ALT> / 1000."

Example

CAS:	"Viper 22, Bat 10 established at Bush, angels 16."

Initiation of attack run

This is an indication from the CAS to the player that CAS has reached the IP described in its nine line and has begun traveling toward the target. There remains approximately one minute until the CAS will call "Wings level," but the time for that call is calculated by the game engine based on the flight model.

Trigger

CAS departs the IP described in its nine line and is traveling toward the target.

Template

CAS:	"<PLAYER_CALLSIGN>, <CALLSIGN_CAS>, IP inbound."

Example

CAS:	"Viper 22, Bat 10, IP inbound."

Ready for attack clearance

This is an indication from the CAS to the player that CAS is ready for an attack clearance, i.e., the CAS is pointed toward the target, its wings are level, and there exists no terrain blocking a straight line drawn from the CAS to the target. In real life, the CAS pilot will make this call when he can see the target. For game mechanics, we need to specify some arbitrary distance at which a pilot *could* see the target. Call that distance 2500 meters.

Trigger

CAS is within 2500 meters of the target, is pointed toward the target, is wings level, and there exists no terrain blocking a straight line drawn from the CAS to the target.

Template

CAS:	"<PLAYER_CALLSIGN>, <CALLSIGN_CAS>, wings level."

Example

CAS:	"Viper 22, Bat 10, wings level."

Egress to stack

This is an indication from the CAS to the player that CAS has completed the attack run, either by dropping ordnance on the target, or by receiving an abort command. In either case, the CAS then flies along its egress vector back to the assigned stack point.

Trigger

CAS has completed the attack run by either receiving a clearance to drop ordnance or by receiving an abort call.

Template

CAS:	"<PLAYER_CALLSIGN>, <CALLSIGN_CAS>, off to the <FOCUS_NINELINE_EGRESS_INIT_CARDINAL>."

Example

CAS:	"Viper 22, Bat 10, off to the east."

FDC MESSAGES

Shot call

This is generated when an indirect fire unit fires a round toward a target. For game purposes, this message is published at 30 seconds prior to TOT for SEAD. For Adjust Fire missions, this message is published 4 minutes after the indirect fire asset sends the MTO.

Trigger

MTO + 4 minutes

Template

FDC:	"<PLAYER_CALLSIGN>, <FDC_CALLSIGN>, shot, over."

Example

CAS:	"Viper 22, R7M, shot, over."

Splash call

This is generated 5 seconds prior to an indirect fire round impact. For game purposes, this message is published at 25 after the shot call for both SEAD and Adjust Fire missions.

Trigger

Shot call + 25 seconds

Template

FDC:	"<PLAYER_CALLSIGN>, <FDC_CALLSIGN>, splash, over."

Example

CAS:	"Viper 22, R7M, splash, over."

ONCOMING FAC(A) MESSAGES

Check in with Air Officer

The oncoming FAC(A), ideally the mission relief for the player, will check in with the Air Officer regardless whether or not the player has terminal control. The Air Officer's response is dependent on whether or not the player has taken terminal control, however, as shown in the template and examples.

Trigger

Oncoming FAC(A) spawns.

Template

Oncoming FAC(A): “<AIR_OFFICER_CALLSIGN>, this is <ONCOMING_FAC(A)_CALLSIGN>, mission number <MISSION_NUM_OCFA>, up as fragged at <CP_SPAWN_OCFA>, cherubs 2 with universal. Playtime (45 - <MINUTES_SINCE_LAUNCH>). Request friendly and enemy situation update.”

Air Officer: *if(playerHasCheckedIn){*

“Roger, <ONCOMING_FAC(A)_CALLSIGN>, copy up as
fragged.”

if(hasTermContPlayer){

“<CALLSIGN_PLAYER>, has terminal control. Contact
him on this push for situation update.”

}else{

“Anchor <CP_SPAWN_OCFA> cherubs 2 and stand by
for situation update. Break, break,
<CALLSIGN_PLAYER>,
<CALLSIGN_AIR_OFFICER>, contact DASC for
routing.

(EXIT SCREEN FORCES PLAYER TO END SESSION)

}

}else{

“Roger <CALLSIGN_OCFA>, copy up as fragged. Push to
<CP_RECON_PLAYER> and stand by for situation update.”

if(there exists at least one friendly artillery unit){

“Be advised”

for(i=0; i<number of artillery units; i++){

“<CALLSIGN_ARTY_FRIENDLY(i)> located at
<GRID_ARTY_FRIENDLY(i)>, gun target line
<GUNLINE_ARTY_FRIENDLY(i)>.”

}

}

(TIME DELAY 10 SECONDS)

“<CALLSIGN_OCFA>, <CALLSIGN_AIR_OFFICER>,

<SITUATION_AIR_OFFICER>"

(EXIT SCREEN FORCES PLAYER TO END SESSION)

}

Example 1: PLAYER HAS TERMINAL CONTROL

Oncoming FAC(A):	"Mongo, this is Viper 20, mission number 3016, up as fragged at Star, cherubs 2 with universal. Playtime 0+45. Request friendly and enemy situation update."
Air Officer:	"Roger, viper 20, copy up as fragged. Viper 18 has terminal control. Contact him on this push for situation update."

Example 2: PLAYER HAS CHECK IN BUT HAS NOT TAKEN TERMINAL CONTROL

Oncoming FAC(A):	"Mongo, this is Viper 20, mission number 3016, up as fragged at Star, cherubs 2 with universal. Playtime 0+45. Request friendly and enemy situation update."
Air Officer:	"Roger, viper 20, copy up as fragged. Anchor Star cherubs 2 and stand by for situation update. Break break, Viper 18, Mongo, contact DASC for routing.

(EXIT SCREEN FORCES PLAYER TO END SESSION)

Example 3: PLAYER HAS NOT CHECKED IN

Oncoming FAC(A):	"Mongo, this is Viper 20, mission number 3016, up as fragged at Star, cherubs 2 with universal. Playtime 0+45. Request friendly and enemy situation update."
Air Officer:	"Roger Viper 20, copy up as fragged. Push to Spider and stand by for situation update. Be advised R7M located at NU 682 187, gun target line 065."

(TIME DELAY 10 SECONDS)

"Viper 20, Mongo, 1st Battalion, 7th Marines was ambushed by a mechanized infantry battalion while in pursuit of the enemy after they retreated from their assault through Noble Pass. Bravo company is encountering heavy resistance in vicinity of grid NU 740 230. They're engaged with the enemy's lead elements in the open. The CO wants to break the enemy lines so Bravo can push through to envelop the ambush forces. Viking 20 is a section of Hornets out of Mina Al Palms; they should be arriving on station any minute for CAS. R7M is in direct support of 1/7 with six guns. My position is 7 clicks southwest of Spider, and I do not have eyes on. I need continuous coverage over the engagement area. You will need to run Viking out of the south. Scouts have sighted 2 x ZSU-23-4 seven clicks northeast of Cloud in the wash behind the enemy front. Map datum is WGS-84; all players on universal. Type one CAS in effect. My FAC, callsign Beaker, is moving up to Bravo’s position and may be ready to direct fires at the end of your playtime. He'll be up TAD-5 and local; I will be monitoring.”

Ready for terminal control

This is generated 5 seconds prior to an indirect fire round impact. For game purposes, this message is published at 25 after the shot call for both SEAD and Adjust Fire missions.

Trigger

Shot

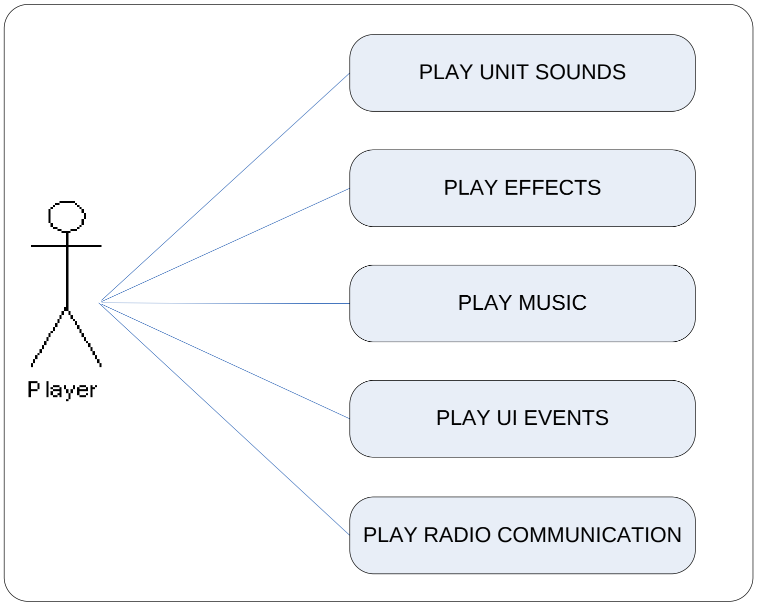
Template

FDC:	“<PLAYER_CALLSIGN>, <FDC_CALLSIGN>, splash, over.”

Example

CAS:	“Viper 22, R7M, splash, over.”

AUDIO SYSTEM



1. Play Unit Sounds
Sounds of tanks, jets, ownship.

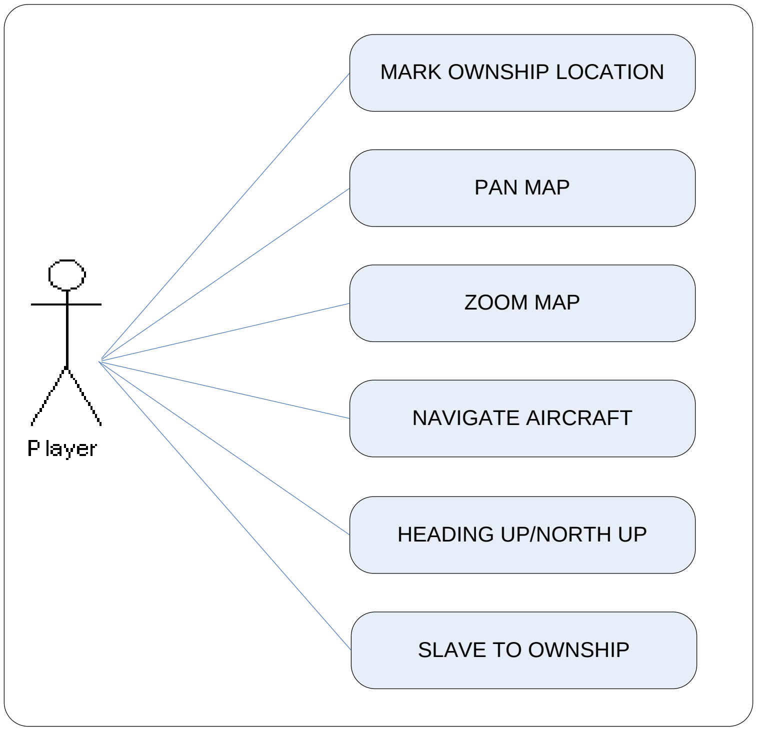
2. Play Special Effect Sounds
One-time sound events like explosions.

3. Play Music

4. Play User Interface Event Sounds
Button clicks.

5. Play Radio Communication Voices

MINI-MAP SYSTEM



The Mini-Map renders the chart map for the current mission. The user can pan and zoom the chart map. Displayed on the chart map are icons representing known units and locations. The user's aircraft path and waypoints are also rendered.

The Mini-Map window is rendered on top of the Out the Window pane and can be moved by left clicking and dragging on the window's top border. The Mini-Map window can be resized by left clicking and dragging on one of the four window corners.

1. Mark Ownship Location

The user can left-click on the Mark Position Button which will display an icon on the MiniMap to denote the position of the ownship at that point and time.

2. Pan Map

The map is panned by the user left-clicking and dragging the mouse in the Mini-map window. The amount of chart panning is equal to the amount the mouse moves. The pan will not exceed the boundaries of the chart map. This will restrict ownship movement to the area the chart map represents.

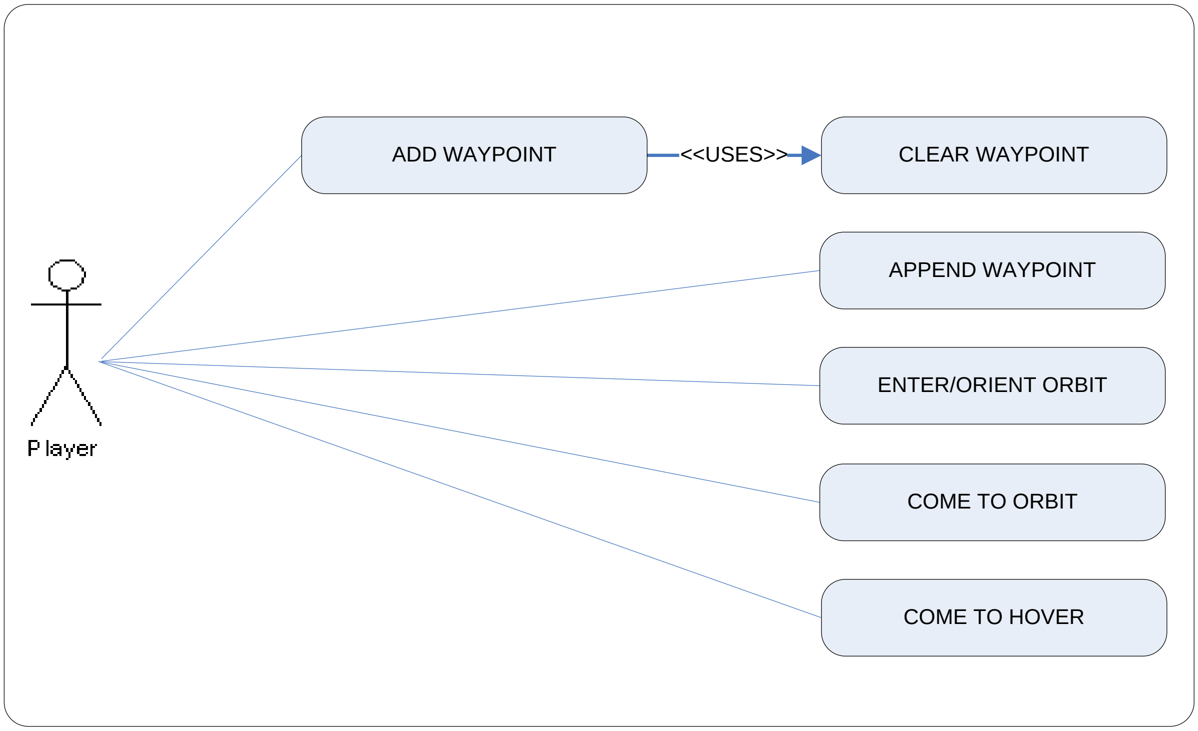
3. Zoom Map
The user can zoom in/out of the chart map by left-clicking and dragging the zoom scroll bar. The most a user can zoom in until the size of the displayed map on the monitor is the same as the size of the real life print. The most a user can zoom out is until the full chart map is in view.

4. Navigate Aircraft (See Aircraft Navigation System)

5. Toggle Heading up / North up
The user can left-click on the Map Orientation button to switch the map orientation from north-up to ownship heading-up.

6. Toggle Slave to Ownship
The user can left-click on the Slave to Ownshp button which will cause the MiniMap to keep the Ownship symbol in the center of the map.

AIRCRAFT NAVIGATION SYSTEM



The user controls the aircraft navigation by adding one or more waypoints to traverse. The aircraft automatically flies to the next waypoint. The aircraft will hover when it reaches the last waypoint. The User has the option to make the aircraft enter a racetrack pattern on the last waypoint entered. The orientation of the racetrack defaults to be inline with the last direction of travel.

1. Add Waypoint
In the Mini-map, the user right clicks on an empty space to add a waypoint to fly to. Adding a new waypoint will remove the existing waypoints.

2. Clear Waypoints
Remove all the existing waypoints.

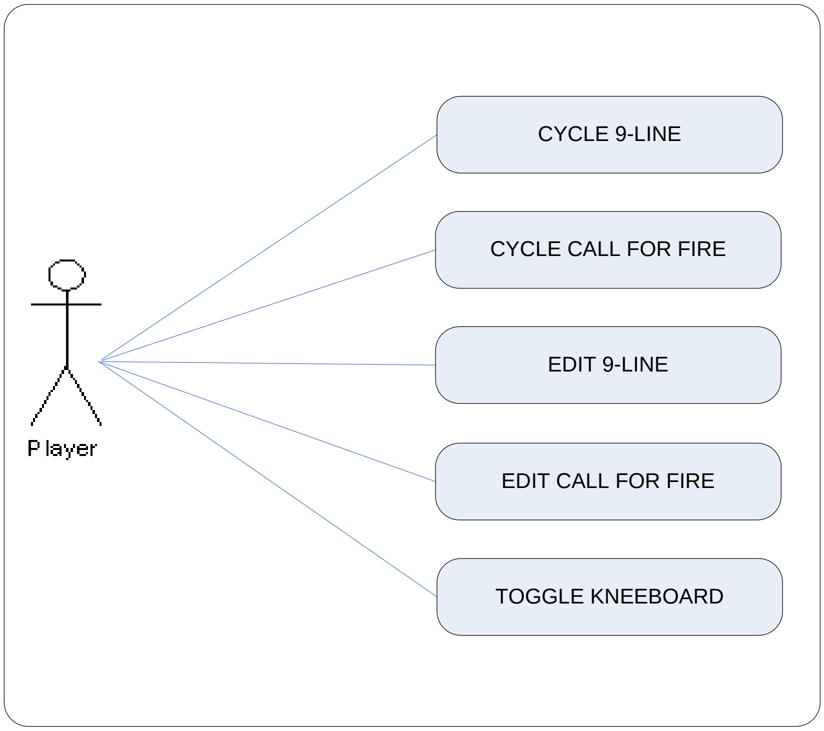
3. Append Waypoint
In the Mini-map, the user shift right clicks on an empty space. The waypoint is added to the end of the list and the route is rendered.

4. Enter and Orient the Racetrack Orbit
In the Mini-map, the User right clicks and drags the cursor. The mouse drag direction will orient the racetrack pattern.

5. Come to Orbit
In the Mini-map, the user left-clicks on the Orbit Button. The aircraft automatically comes into a racetrack pattern, with the longitude of racetrack aligned with the last direction of travel.

6. Come to Hover
In the Mini-map, the user left clicks on the Hover Button. The aircraft automatically comes to a hover aligned to the last direction of travel.

KNEEBOARD SYSTEM



1. Cycle through the 9-lines
The user can cycle through the stack of existing 9-lines by left clicking on the Next 9-line and Previous 9-line Buttons.

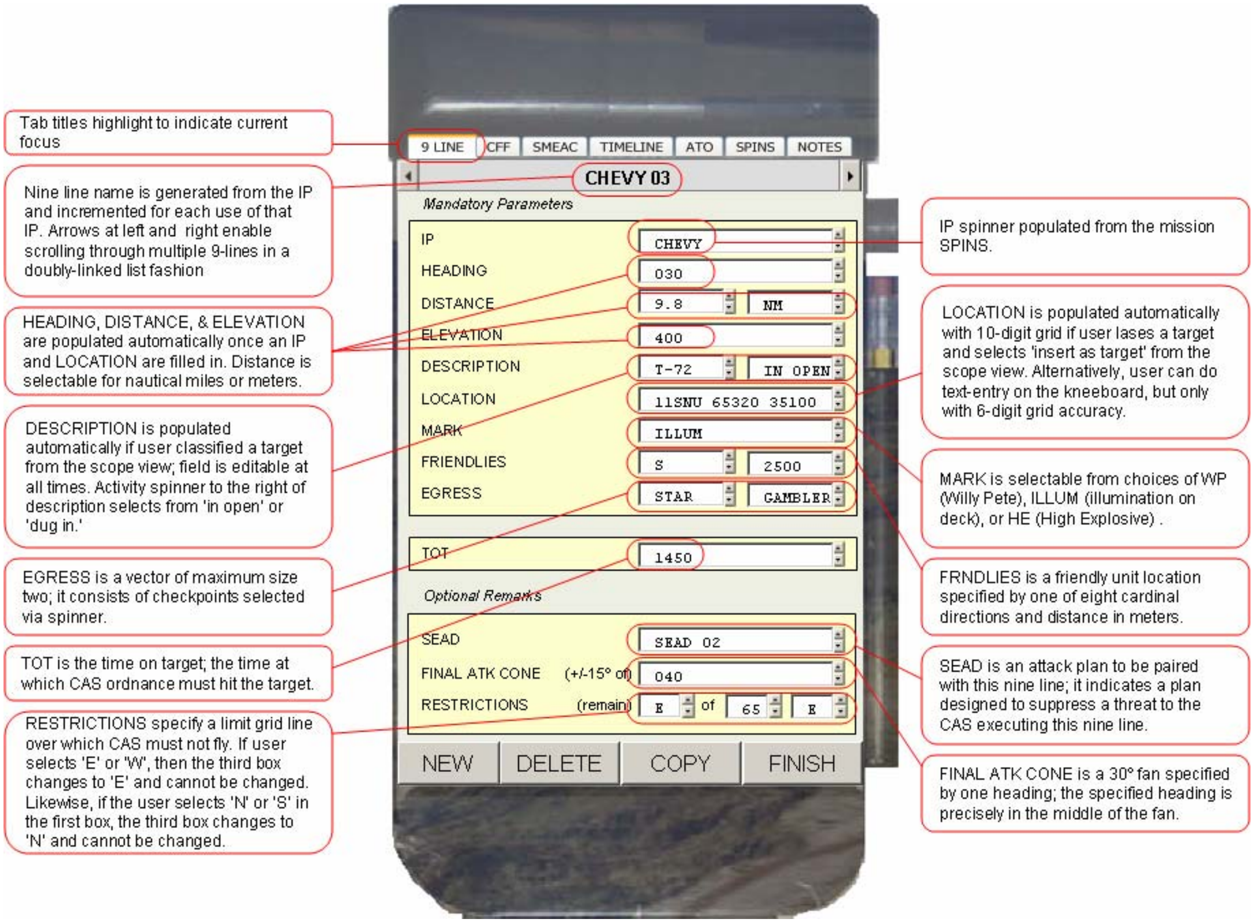
2. Cycle through the Call for Fire
The user can cycle through the stack of existing 9-lines by left clicking on the Next 9-line and Previous 9-line Buttons.

3. Edit a 9-line

4. Edit a Call for Fire

5. Toggle the Display of the Kneeboard
The user can toggle the display of the Kneeboard by pressing the Kneeboard Toggle Button on the Main View. The Kneeboard should remain in the right side of the window.

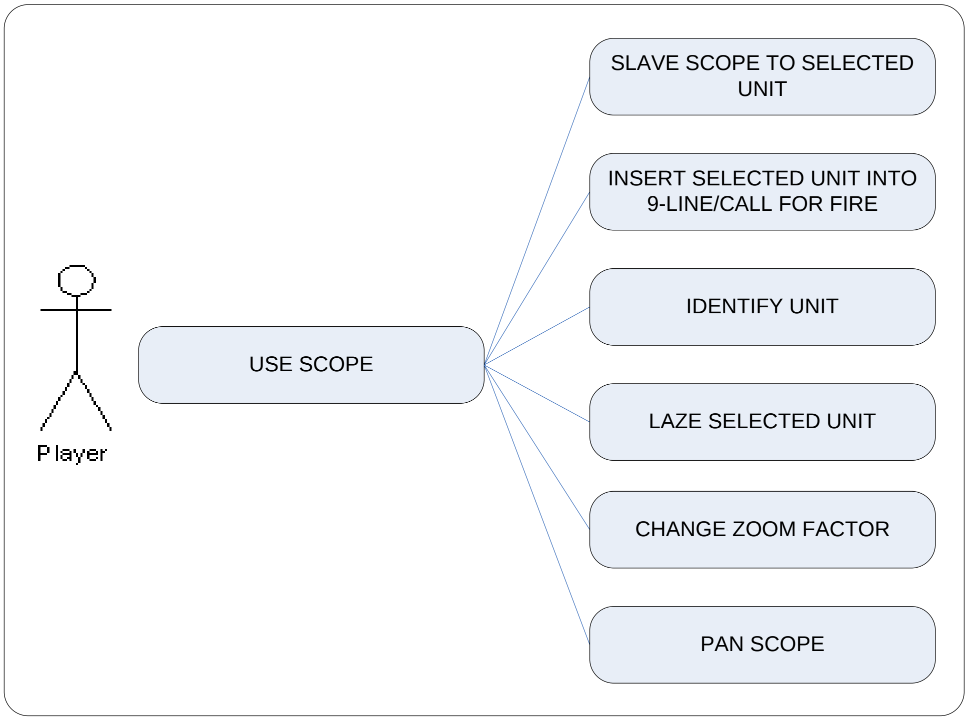
NINE LINES



FIRE MISSIONS, ATO, SPINS, NOTES, TIMELINE

Implemented in a similar manner as the nine line tab, the fire mission tab allows formulation of orders to indirect fire assets. The ATO and SPINS tabs provide a view of those mission documents, and the Notes tab allows the user to record information in free-form. The Timeline gives a visual depiction of attack orders past, projected, and in progress.

SCOPE SYSTEM



Slave Scope to Tracked Unit
The user can Track a unit by left-clicking on the Track button when a unit is under the crosshairs of the Scope. Once tracked, the scope will automatically slave to keep the unit in the center of the screen. To un-track, the user can left-click on the Track button a second time.

Insert Selected Unit into 9-line

Identify Selected Unit
The user selects the unit (see Select Unit). The user then fills out the proper information in the Scope View and left clicks the "OK" button.

The identified unit will then be displayed on the Chart Map and Mini-Map using the unit type the user entered.

Laze Selected Units

Change Zoom Factor
Two increments: 2x mag (30 deg FOV) and 13x mag (4.6 deg FOV)

Pan Scope

The user left clicks and drags in the Scope view. The view is rotated and pitched the same amount the cursor is dragged. The view adjustments are limited to +/- 90 degrees horizontal and +25/-50 degrees vertical.

If a unit is currently selected, a Pan operation will (…)

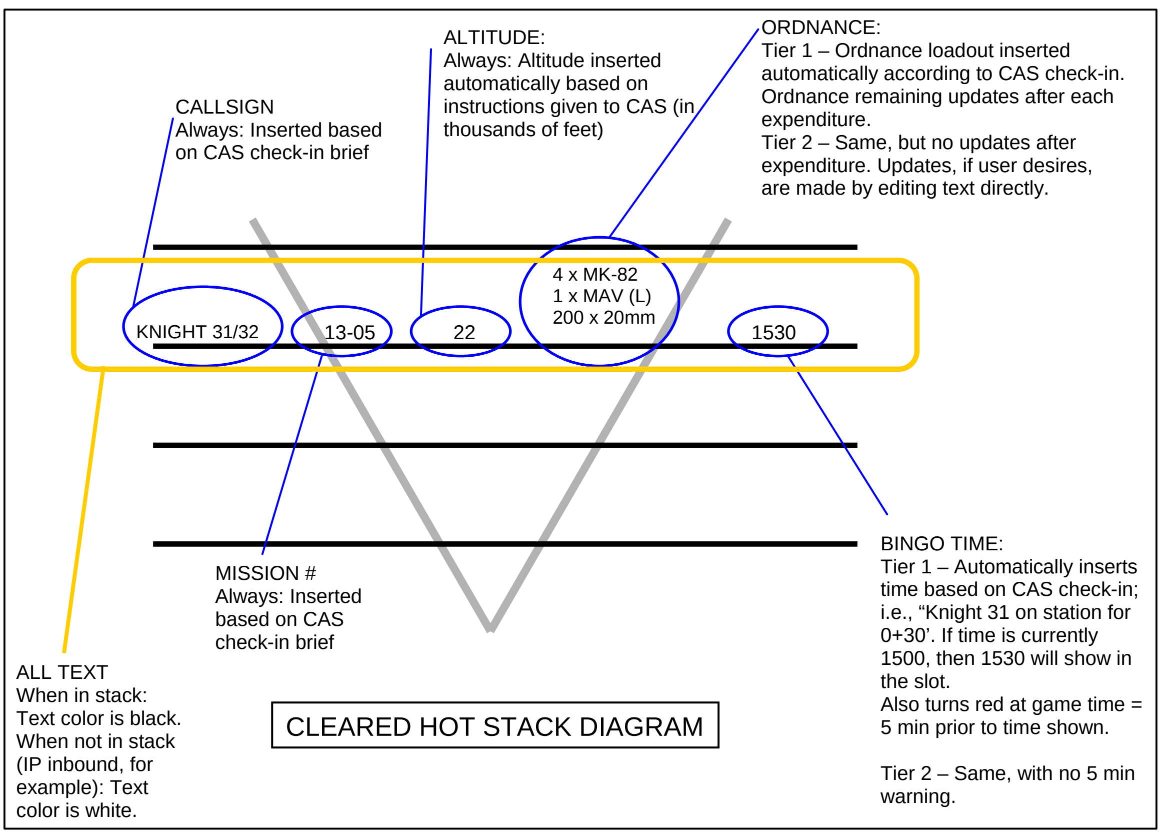
STACK DIAGRAM SYSTEM

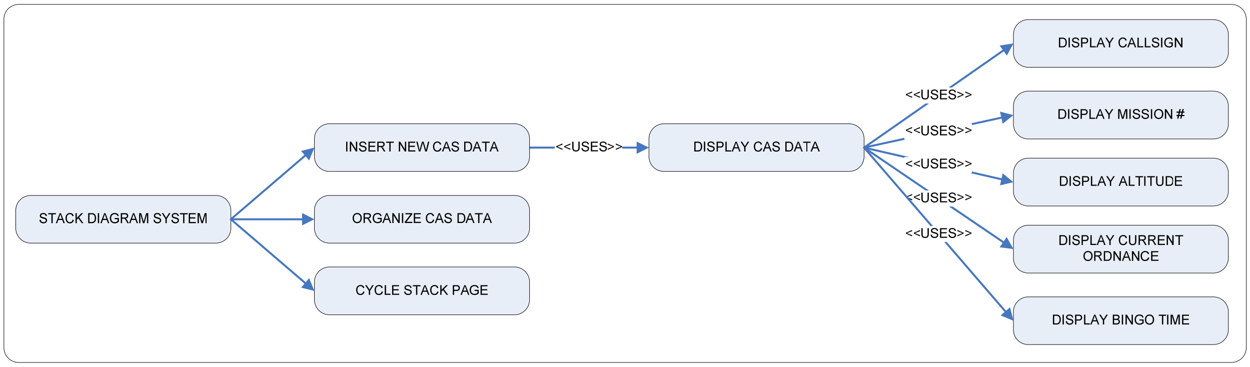
The stack diagram is a graphical depiction of how the FAC has arranged the CAS assets. There are a total of three (3) stacks diagrams available to the user and are cycled by left clicking on the next and previous stack buttons. The three available stack points are a subset of the Check Points listed in the Scenario (in the ATO briefing).

Upon scenario load, the stack diagram is blank and the next/previous buttons are disabled. Once the user assigns a CAS asset to a stack position, the CAS information is added to the diagram, starting from the bottom line.

As additional CAS assets get added to a stack point, the lines in the diagram sorted based on the assigned altitude with the highest altitude on the top. If more than one CAS is assigned the same altitude, the oldest assigned CAS is moved higher in the diagram.

When the CAS leaves the stack point, its data changes from black to white.
When a CAS Returns to Base (RTB), its associated data is removed from the stack diagram.
The ordnance gets automatically decremented when the CAS fires a weapon.





Insert New CAS Data
The user assigns a CAS group to a stack point, which causes the CAS' data to be entered into the Stack Diagram.

Organize CAS Data
The CAS' data line is positioned according to the altitude the CAS was assigned to. The CAS data is entered from the bottom first and sorted based on altitude (lowest altitude is at the bottom of the diagram).

Cycle Stack Page
The user left clicks on the next/prev stack buttons which causes the Stack Diagram to cycle through the 3 stack point pages.

Display CAS Data

Display Callsign
The CAS callsign is displayed as text. Data comes from the scenario file.

Display Mission #
The mission number is displayed as text. Data comes from the scenario file.

Display Altitude
The assigned altitude of the CAS is displayed as text as thousands of feet AGL.

Display Current Ordnance
The CAS' current ordnance count is displayed as text and decremented whenever the CAS fires a weapon.

Display Time on Station
The CAS' time on station is displayed as text using the scenario clock. If the CAS checks in with "+30" and the scenario time is 1500, the stack diagram will display "1530".

If the time on station is within 5 minutes of the scenario clock, the time on station text will be highlighted.

RADIO DIALOG VARIABLES

VARIABLE	SOURCE	CLEARED HOT ALPHA RELEASE VALUE
CALLSIGN_PLAYER	Main menu selection	TBD
CALLSIGN_AIR_OFFICER	MEA	Mongo
POSITION_ROUGH_AIR_OFFICER	MEA	seven clicks southwest of Spider
CALLSIGN_FAC	MEA	Beaker
ASSIGNED_TERMINAL_POINT	ATO(MEA)	Star
AIRCRAFT_ALTITUDE	CONSTANT	200
MINUTES_SINCE_LAUNCH	GE1	TBD
PLAYER_BINGO	GE8	TBD
RECCE_CP	MEA	Spider
FRIENDLY_SUB_UNIT_SUPPORTED_VERBOSE	MEA	Bravo company 1/7
FRIENDLY_SUB_UNIT_SUPPORTED_ABBREV	MEA	Bravo
FRIENDLY_HIGHER_UNIT_SUPPORTED_VERBOSE	MEA	1st Battalion, 7th Marines
FRIENDLY_ HIGHER_UNIT_SUPPORTED_ABBREV	MEA	1/7
FRIENDLY_UNIT_SUPPORTED_GRID	MEA	NU 74 23
FRIENDLY_AIR_UNIT_ASSIGNED_CP(i)		
SELECTED_FRIENDLY_UNIT_GRID(i)		
SELECTED_FRIENDLY_UNIT_NAME_ABBREV(i)		
LAST_ENEMY_UNIT_ATTACKED_DESC		
LAST_ENEMY_UNIT_ATTACKED_GRID		
CAS_MISSION_CAPABILITY		
COMMANDER_INTENT	MEA	The CO wants to break the enemy lines so Bravo can push through to envelop the ambush forces.
SITUATION_AIR_OFFICER	MEA	0: “<FRIENDLY_HIGHER_UNIT_SUPPORTED_VERBOSE> was ambushed by a mechanized infantry battalion while in pursuit of Chipotlean forces after the enemy retreated from their assault through Noble Pass. <FRIENDLY_SUB_UNIT_SUPPORTED_ABBREV> is encountering heavy resistance in vicinity of grid <FRIENDLY_SUB_UNIT_SUPPORTED_GRID>. They're engaged with the enemy’s lead elements in the open. <COMMANDER_INTENT>. <FRIENDLY_AIR_UNIT_CALLSIGN(0)> is a <FRIENDLY_AIR_UNIT_NO_ACFT_COMMON(0)> of <FRIENDLY_AIR_UNIT_TYPE_ACFT_COMMON(0)> out of Mina Al Palms; they should be arriving on station any minute for CAS. <FRIENDLY_ARTY_UNIT_NAME(0)> is in direct support of <FRIENDLY_HIGHER_UNIT_SUPPORTED_ABBREV> with six guns.” 1: “My position is <AIR_OFFICER_POS>, and I do not have eyes on. I need continuous coverage over the engagement area. You will need to run <FRIENDLY_AIR_UNIT_CALLSIGN(0)> out of the south. Scouts have sighted <ENEMY_ADA_SIT_BRIEF_DESC>.” 2: “Map datum is WGS-84; all players on universal. Clear all missions on TACP local; type one CAS in effect. My FAC, callsign <GROUND_FAC>, is moving up to <FRIENDLY_SUB_UNIT_SUPPORTED_ABBREV> ‘s position and may be ready to direct fires at the end of your playtime. He'll be up TAD-1 and local; I will be monitoring.” 3: “<CALLSIGN_AIR_OFFICER>, <CALLSIGN_PLAYER>, copy all. Switching TAD-1,

VARIABLE	SOURCE	CLEARED HOT ALPHA RELEASE VALUE
		out.”
FRIENDLY_ARTY_UNIT_NAME(i)	MEA	0: R7M
FRIENDLY_ARTY_UNIT_GRID(i)	MEA	0: NU 682 187
FRIENDLY_ARTY_UNIT_GTL(i)	MEA	0: 065º
FRIENDLY_AIR_UNIT_CALLSIGN(i)	MEA	0: Viking 20 1: Wake 30 2: Bat 10
NINELINE(i)	Kneeboard (PG)	0 - n: TBD
NINELINE_AIR_UNIT(i)	GE6	0: Viking 20 1: Wake 30 3: B at 10
NINELINE_IP(i)	Kneeboard (PG)	0 - n: TBD
NINELINE_TGT_DESC(i)	Kneeboard (PG)	0 - n: TBD
NINELINE_TGT_LOC(i)	Kneeboard (PG)	0 - n: TBD
NINELINE_EGRESS(i)	Kneeboard (PG)	0 - n: TBD
NINELINE_TOT(i)	Kneeboard (PG)	0 - n: TBD
NINELINE_MARKING_UNIT(i)	GE7	0 - n: TBD
ENEMY_ADA_SIT_BRIEF_DESC	MEA	Two ZSU-23-4 seven clicks northeast of Cloud in the wash behind the enemy front
ENEMY_ADA_ALIVE_DESC(i)	GE2	0 – n: TBD
ENEMY_ADA_ALIVE_GRID(i)	GE2	0 - n: TBD
ENEMY_GROUND_ALIVE_DESC(i)	GE2	0 – n: TBD
ENEMY_GROUND_ALIVE_GRID(i)	GE2	0 - n: TBD
FRIENDLY_GROUND_ALIVE_DESC(i)	GE2	0 - n: TBD
FRIENDLY_GROUND_ALIVE_GRID(i)	GE2	0 - n: TBD
FRIENDLY_AIR_UNIT_MISSION_NO(i)	MEA	0: (TODAY’S DATE) + 07 1: (TODAY’S DATE) + 09 2: (TODAY’S DATE) + 11 Example: if today is the 16th day of the month, then the two missions are 1607 and 1609
PLAYER_MISSION_NO	MEA	(TODAY’S DATE) + 15
FRIENDLY_AIR_UNIT_QTY_DIGITS(i)	MEA	0: 2 1: 2 2: 2
FRIENDLY_AIR_UNIT_QTY_DESC(i)	MEA	0: section 1: section 2: section
FRIENDLY_AIR_UNIT_TYPE_ACFT_FORMAL(i)	MEA	0: F/A-18D 1: AV-8B 2: F/A-18D
FRIENDLY_AIR_UNIT_TYPE_ACFT_COMMON(i)	MEA	0: Hornets 1: Harriers 2: Hornets
FRIENDLY_AIR_UNIT_ORD_REMAINING(i)	MEA/GE3	0: <F/A-18D_STANDARD_LOADOUT_0>, less ordnance expended 1: <AV-8B_STANDARD_LOADOUT_0>, less ordnance expended 2: <F/A-18D_STANDARD_LOADOUT_0>, less ordnance expended
FRIENDLY_AIR_UNIT_CURRENT_POS(i)	GE4	0 - n: TBD
FRIENDLY_AIR_UNIT_CURRENT_ALT(i)	GE4	0 - n: TBD
FRIENDLY_AIR_UNIT_BINGO_TIME(i)	Stack Diagram	0 - n: TBD
FRIENDLY_ AIR_UNIT_RADIO_NET(i)	MEA	0: TAD-5 1: TAD-5 2: TAD-5
FRIENDLY_ AIR_UNIT_CONTROLLER	CONSTANT	<CALLSIGN_PLAYER>
CURRENT_TIME	MEA/GE5	System clock
PLAYER	Main menu selection	TBD
AIR_OFFICER	MEA	Mongo

VARIABLE	SOURCE	CLEARED HOT ALPHA RELEASE VALUE
FRIENDLY_AIR_UNIT_CALLSIGN(i)	MEA	0: Viking 20 1: Wake 30 2: Bat 10
<CAS_ASSIGNED_STACK_CP>	Mandatory entry. Implemented by “spinner”.	TBD
<CAS_ALT>	Mandatory entry. Implemented by “slider”.	TBD
<ORIENT>	Optional item. Implemented by radio button.	TBD
<ADVISE>	Optional item. Implemented by “spinner” with blank set as default.	TBD

GE NOTES:

1 – Minutes since mission begins for player according to system clock.

2 – Still alive units cannot be predetermined; the destruction of units is dependent on player actions. However, starting unit numbers, types, and positions are produced via the mission editor.

3 – Starting ordnance is specified by the mission editor; remaining ordnance is wholly dependent on how many attacks the player has directed. NOTE: Got the feelers out for typical Hornet/Harrier loadouts from a couple of pals at NPS. This loadout will be what is referenced by the variables AV-8B_STANDARD_LOADOUT_0 and F-18D_STANDARD_LOADOUT_0 in the table above. Will have that data this week (14 Nov).

4 – Where the aircraft ends up at the time of BHO is driven by player behavior.

5 – System time is used as the one-to-one scale of minutes passing in the game to minutes passing in real life, but absolute times are not the same. For example, the scenario start time is specified by the mission editor as 1600, but of course the player may start the Cleared Hot program at any time of the day. After the player has played the game for 15 real-life minutes, the value of CURRENT_TIME will be 1615.

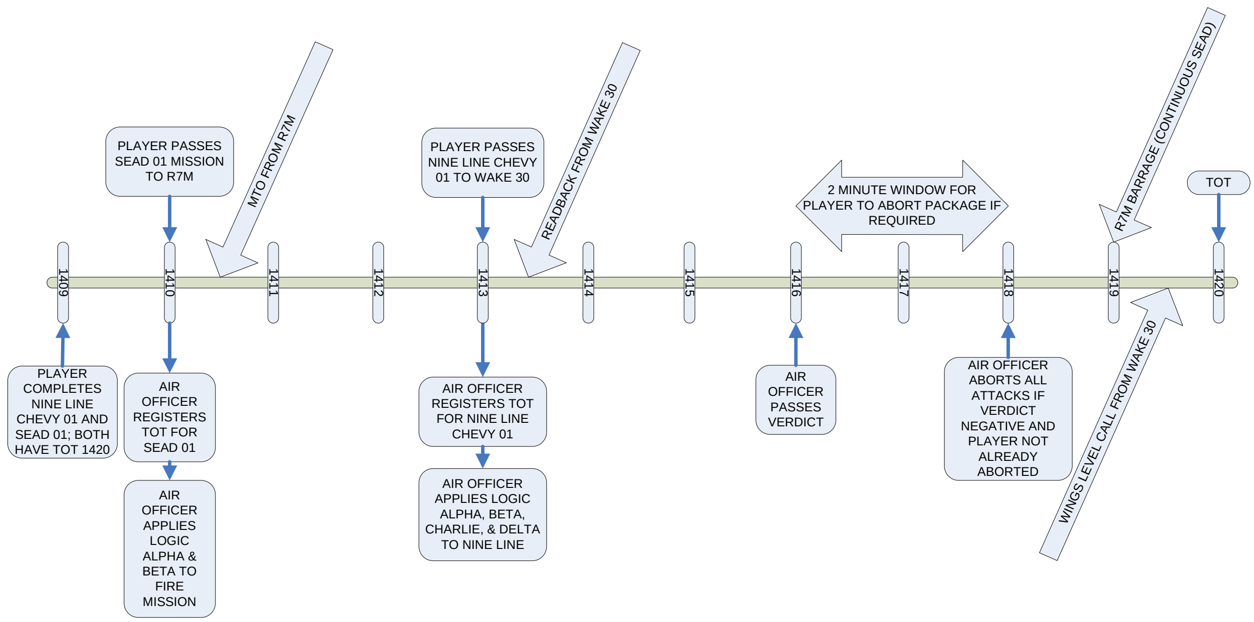
6 – A single nine-line may be passed to several air units. Consequently, once player passes a nine-line with a TOT to an air unit, the air unit and the nine-line need to be linked/associated.

7 – Line 7 of nine-line is mark type (e.g. smoke, laser, etc). Player needs a venue/method with which to identify who is actually providing the mark (i.e. indirect fire unit or ownship). The marking unit and the nine-line need to be linked/associated.

8 – Need to project system clock time for when MINUTES_SINCE_LAUNCH == 90.

LOGIC SUBROUTINES

ATTACK PACKAGE APPROVAL AND EXECUTION TIMELINE



LOGIC FILTERS

ALPHA	APPROVE: TARGET <DANGER_CLOSE_LENGTH> RADIUS DOES NOT CONTAIN FRIENDLY UNITS
BETA	APPROVE: IF(SEAD MISSION){ TOT >= <CURRENT_TIME> + 6 MINUTES }, IF(NINE LINE){ TOT >= <CURRENT_TIME> + 4 MINUTES
CHARLIE	APPROVE: (TARGET <THREAT_DISTANCE> DOES NOT CONTAIN ENEMY ADA) or (TARGET <THREAT_DISTANCE> DOES CONTAIN ENEMY ADA *and* SEAD TARGET == ADA)
DELTA	APPROVE: CAS PATH DOES NOT INTERSECT VERTICAL PLANE THAT INCLUDES SEAD UNIT GRID AND SEAD TARGET GRID

APPENDIX

Call for Fire
Options available to use:

Warning order
Type of mission (Adjust Fire, Fire for Effect, SEAD)
Method of target location (Grid)

Target Location
Grid Value

Target Description
What the target is (trucks, troops, etc.)
What the target is doing (digging in, attacking, etc.)
Amount of cover (in the open, in bunkers, etc.)
Number of elements (3 trucks, squad, etc.)

Method of Engagement
Ammunition Request (HE, WP)

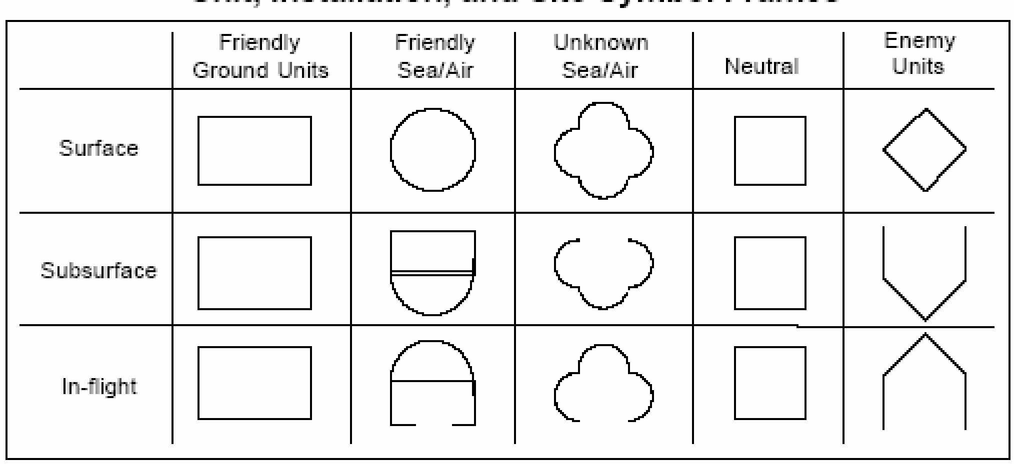
Method of Fire and Control
When ready
At my command
TOT

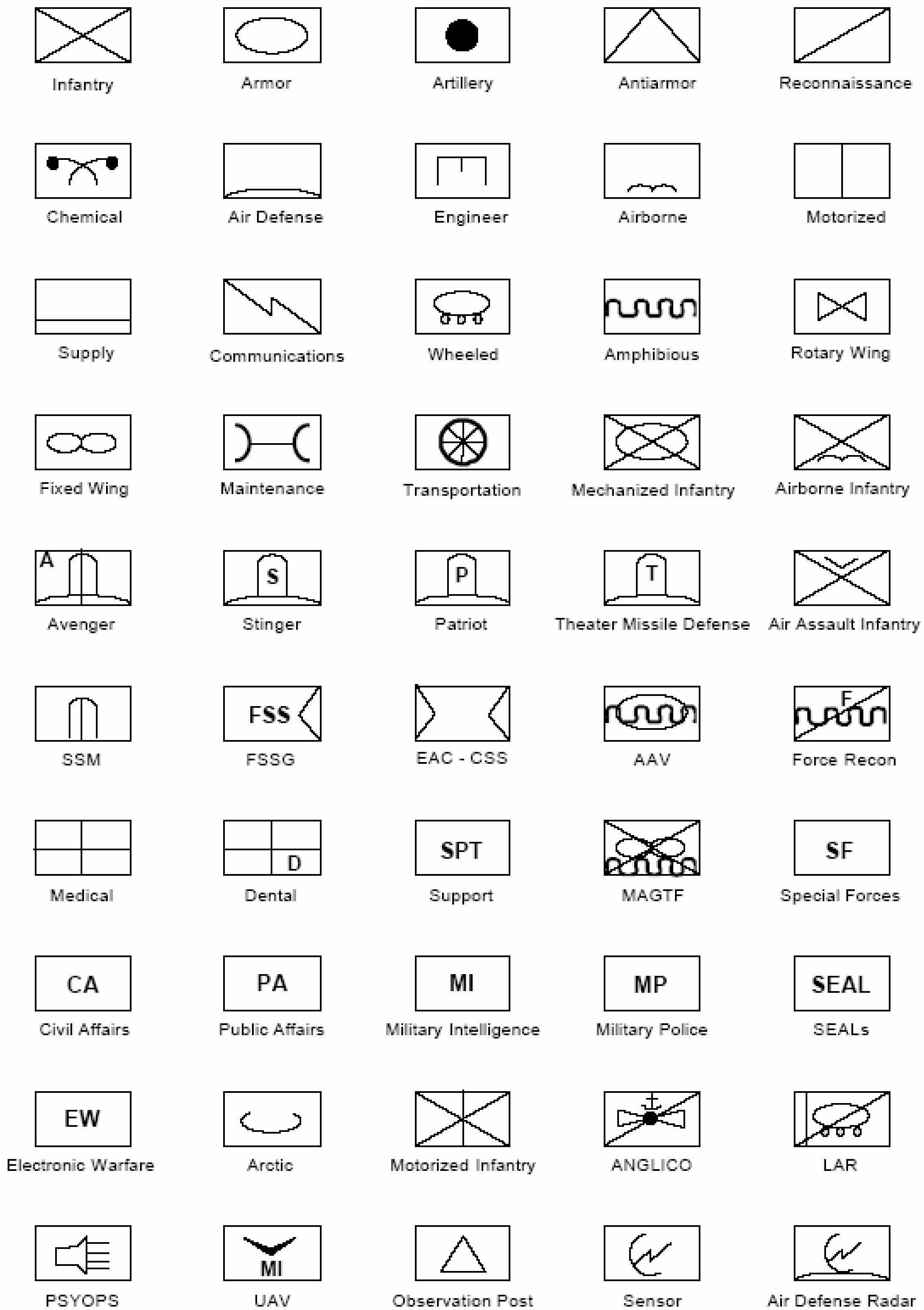
Mini-map Symbology

Size Indicator	Meaning
■	Installation
Ø	Team / Crew
●	Squad
● ●	Section
● ● ●	Platoon / Detachment
I	Company / Battery / Troop
I I	Battalion / Squadron
I I I	Regiment / Group
X	Brigade
X X	Division
X X X	Corps
X X X X	Army
X X X X X	Army Group / Front
X X X X X X	Region

Unit Size and Installation Indicator

Unit, Installation, and Site Symbol Frames




Infantry
Armor
Artillery
Antiarmor
Reconnaissance
Chemical
Air Defense
Engineer
Airborne
Motorized
Supply
Communications
Wheeled
Amphibious
Rotary Wing
Fixed Wing
Maintenance
Transportation
Mechanized Infantry
Airborne Infantry
A
Avenger
S
Stinger
P
Patriot
T
Theater Missile Defense
Air Assault Infantry
SSM
FSS
FSSG
EAC - CSS
AAV
F
Force Recon
Medical
D
Dental
SPT
Support
MAGTF
SF
Special Forces
CA
Civil Affairs
PA
Public Affairs
MI
Military Intelligence
MP
Military Police
SEAL
SEALs
EW
Electronic Warfare
Arctic
Motorized Infantry
ANGLICO
LAR
PSYOPS
MI
UAV
Observation Post
Sensor
Air Defense Radar

5-10

REVISION HISTORY

10/25/05 We changed the Unit Selection process to only happen from the Scope View. The only reason to select a unit is to identify it. You can't identify it if you can't see it. You can't see it unless you use the Scope (usually). The unit identification GUI is on the Scope view as well.

As per discussion with Lakey, King, Grant, and Johnson, the only possible way to select a unit, is by clicking on the 3D object in the scope.

The Scope has angle restrictions and zoom functionality noted as per Lakey.

The user can now pan the Scope manually. This is needed since the user can only select the unit if it's in the FOV of the scope.

The Out the Window adjustment is limited per Lakey (roughly related to what the navigator can see from his position in the cockpit).

10/26/05 Replaced "6-line" references to "Call for Fire" because CFF is the proper terminology for communicating with the indirect fire unit.

10/27/05 Added Display Mini-map Use Case; added Mini-Map Use Case.

10/31/05 Expanded Air Officer radio dialog options per Lakey/King discussion. The Viewport Adjustment was modified to allow the aircraft to adjust its orientation once the viewport limits are reached. Defined what the kneeboard needs to display. Defined the Call for Fire options in the Appendix section.

10/31/05 Added Mini-map symbology information.

11/1/05 Better described the functionality of the Radar. Updated main game screen definition. Added screenshots of Display Use cases.

11/1/05 Chart map displays a 1:50k map (per Lakey)

11/1/05 Added additional information regarding the Kneeboard.

11/9/05 Augmented the Radar View Display system (per Lakey/King).

11/21/05 Changed Radar View to have a max range of 15km (per Lakey/King).

1/17/06 Added the Stack Diagram System.

1/24/06 Cleaned up a few things based on current implementation.

1/24/06 Added MiniMap tracking options

1/25/06 Added Mark feature for the MiniMap. Added Come to Orbit Button in the MiniMap

2/13/06 Added graphic depiction for the comm dialogue interface; provided description of functionality.

2/14/06 Added narrative example and Hold functionality for CAS dialog.

2/22/06 Added 9-line comm window variable selection methods; added comm dialogue window functionality.

2/27/06 Added description for CAS unit comm acknowledgement routine.

3/1/06 Uploaded all new unit comm diagrams to accurately reflect design decisions made over last 3 weeks.

3/2/06 Added graphics for kneeboard; more on all kneeboard tabs coming soon.

3/6/06 Created full graphic with list population parameters for kneeboard 9-line tab.

3/7/06 Removed reference to the Browse Unit Selector. Added View Mark to the Minimap Display. Optimized the CAS communication images.

3/7/06 Changed main game screen and breakdown.

3/13/06 Defined the rules for what/when icons get displayed on the MiniMap and Radar. Removed reference to "selecting a unit" from the Scope View system and replaced with "tracking a unit".

3/15/06 Added portions of Air Officer and CAS dialogue data, to include templates for conversation, conversation variable sources, and example dialogues.

3/28/06 Revised structure of communication bubble descriptions; implemented standard way of describing functionality for each communication section: GRAPH, CONTEXT OF MESSAGES, LIST OF VARIABLES, TEMPLATE FOR VARIABLE USE, and EXAMPLE OF COMMUNICATION GRAPH.

3/29/06 Created all new graphs with Visio for use cases. This will enable on-the-fly edits of this document during planning sessions. Prior graphs were instanced in this document as non-editable pictures. Cleaned up the graphs where design decisions had made certain elements void.

3/30/06 Extensive redesign of this document with respect to the way communication branches are explained. Should make for easier digesting. TOC updated to reflect actual section placement. All game variables are now located in their own section (that table was getting too big).

4/03/06 Finished revamp of communication branches organization.

4/04/06 Added a new section entitled 'System Generated Messages' that details the unsolicited messages that appear to the player from various game agents. Also made some headway with the player-initiated messages. Chronologically, we're up to the point that the player submits an attack package for approval to the Air Officer.

4/05/06 Small edits on system variable list; nothing significant.

4/05/06 Made preliminary templates for player communications with the FDC.

4/12/06 Redesigned the nine line tab on the kneeboard. New design reflects the incorporation of remarks section in an attack plan.

4/17/06 Continued progress on Call For Fire templates and examples.

4/18/06 Finished Call For Fire templates and examples, added FDC system messages.

4/19/06 Comm dialogues complete with exception of Battle Handover. Added Attack Package execution timeline (with logic gate explanations) to appendices.

4/20/06 Comm dialogues almost done; only remaining are two unsolicited messages: oncoming FAC(A) prompt for situation update and Oncoming FAC(A) request for terminal control.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Marine Corps Representative
   Naval Postgraduate School
   Monterey, California

4. Director, Training and Education, MCCDC, Code C46
   Quantico, Virginia

5. Director, Marine Corps Research Center, MCCDC, Code C40RC
   Quantico, Virginia

6. Marine Corps Tactical Systems Support Activity, (Attn: Operations Officer
   Camp Pendleton, California