AFRL-RH-WP-TR-2008-0079

Developing a Corrective Action Simulator To Support Decision Making Research and Training

Jeffrey A. Doyal
Michael G. Sargent
Roger L. Overdorf
Robert S. McClure
Science Applications International Corp.
4031 Colonel Glenn Hwy
Beavercreek OH 45431

Michael W. Haas
Human Effectiveness Directorate
Warfighter Interface Division
Wright-Patterson AFB OH 45433-7022

May 2008

Final Report from the period November 2006 to June 2008

**Air Force Research Laboratory
Human Effectiveness Directorate
Warfighter Interface Division
Collaborative Interfaces Branch
Wright-Patterson AFB OH 45433**

NOTICE AND SIGNATURE PAGE

TECHNICAL REVIEW AND APPROVAL

AFRL-RH-WP-TP-2008-0079

THIS TECHNICAL REPORT HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION.

FOR THE DIRECTOR

//**signed//**
Michael Haas
Program Manager
System Control Interfaces Branch

//signed//
Daniel G. Goddard
Chief, Warfighter Interfaces Division
Human Effectiveness Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
27 May 2008

2. REPORT TYPE
Final

3. DATES COVERED *(From - To)*
November 2006 – June 2008

4. TITLE AND SUBTITLE
Developing a Corrective Action Simulator to Support Decision Making Research and Training

5a. CONTRACT NUMBER
FA8650-06-C-6750

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
63231F

5d. PROJECT NUMBER
2830

5e. TASK NUMBER
30

5f. WORK UNIT NUMBER
17

6. AUTHOR(S)
Jeffrey A. Doyal*
Michael G. Sargent*
Roger L. Overdorf*
Robert S. McClure*
Michael W. Haas**

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Science Applications International Corp. *
4031 Colonel Glenn Hwy
Beavercreek OH 45431

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Materiel Command **
Air Force Research Laboratory
Human Effectivenes Directorate
Warfighter Interface Division
Collaborative Interfaces Branch
Wright-Patterson AFB OH 45433-7022

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/RHCP

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-RH-WP-TR-2008-0079

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
88 ABW PA Cleared 06/12/08, WPAFB-08-3703.

14. ABSTRACT
The Air Force Research Laboratory, in an effort to identify and evolve human performance modeling tools and their application to human performance research and training, developed a conceptual prototype of a corrective action simulator (CAS). Applied to an AWACS environment, this prototype CAS system employs task-network-based human performance modeling, 3-D visualization including avatars, speech recognition and synthesis, virtual workstations, and computer-based training components to create an immersive environment in which a live operator in the role of Senior Director can observe and interact with synthetic Weapons Directors. The live operator can observe the synthetic Weapons Directors performing their designated functions and issue verbal corrective action when he/she observes an error being made. The scenario represented reflects a real-world event in which corrective action was required to avoid mission failure. Such an immersive system portraying complex, real world events, provides an ideal environment for the study and training of naturalistic decision making. This report provides an overview of the prototype CAS system development process, including lessons learned, and a brief follow-on effort to create avatars tailored to resemble specific individuals.

15. SUBJECT TERMS
AWACS, human performance model, Corrective Action Simulator, naturalistic decision making, human behavior representation, avatar

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
U	U	U	SAR	71	Dr. Michael W. Haas
					19b. TELEPHONE NUMBER *(include area code)*

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

This page intentionally left blank.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

PREFACE

This effort was conducted under contract number FA8650-06-C-6750 with the Collaborative Interfaces Branch of the Air Force Research Laboratory Human Effectiveness Directorate's Warfighter Interface Division (AFRL/RHCP) at Wright-Patterson Air Force Base, Ohio 45433-7022, for the period November 2006 to April 2008. Science Applications International Corporation (SAIC), 4031 Colonel Glenn Highway, Beavercreek, Ohio 45431-7753 was the contractor. Dr. Michael W. Haas (AFRL/RHCP) was the Program and Contract Monitor as well as the Technical Advisor for this effort.

THIS PAGE INTENTIONALLY LEFT BLANK

INTRODUCTION

Over the past decade, AFRL/RH has been exploring modeling and simulation technologies and their application with respect to achieving realistic representations of human behavior and performance. The development of such human performance models (HPMs) and the technologies that allow them to be easily integrated with other simulations can support the advancement of human/system performance across a broad range of applications including simulation-based acquisition, studies and analysis, and training. The Corrective Action Simulator (CAS) project, led by AFRL/RHCP, builds upon earlier Air Force human performance modeling advancement efforts, focusing on enhancing the abilities of HPMs to support more immersive, interactive simulation environments.

Background

Since the Department of Defense declared the development of "Authoritative representations of human behavior" as a key objective in the Modeling and Simulation Master Plan (DOD 5000.59-P, 1995), AFRL/RH has focused on the study and development of improved human behavior representation (HBR)[1] and human performance modeling technologies and the ability to employ HBRs and HPMs to address a variety of human systems engineering and human system performance issues.

[1] The terms "human behavior representation" and "human performance model" are not synonymous, however, in practice they can sometimes be difficult to differentiate. Models of human behavior and the underlying human cognition result in an observable performance. Similarly, human performance models are often based on underlying assumptions about behavior, and in some cases, cognition. For the purposes of this paper, the terms can be used interchangeably.

Most of these efforts have focused on evaluating model performance and/or extending the capabilities of human modeling architectures.

For example, AFRL/RH's Agent-based Modeling and Behavior Representation (AMBR) program has performed a series of investigations examining the ability of HBR architectures to support accurate modeling of human performance across various complex tasks. Modelers, using different modeling tools, were given descriptions of the same task and instructed to develop an HPM that represents a human performing the task. The resulting HPM behavior was then compared to actual human behavior on the task in an effort to judge the effectiveness of the modeling tools. These evaluations have not only allowed researchers to identify the relative strengths and weakness of the architectures evaluated, but also have provided insights that help the HBR architecture developers extend and improve the tools along the way (Gluck, Pew, & Young, 2005).

Another major effort in support of human performance modeling was AFRL/RH's Combat Automation Requirements Testbed (CART) program. The CART program focused on extending human performance modeling technology to allow human operator models to interact with system and environment models in an effort to link human performance to mission effectiveness (Brett, Doyal, Malek, Martin, Hoagland, & Anesgart, 2002). Under the CART program, the Improved Performance Research Integration Tool (IMPRINT) modeling environment was enhanced to support external communications via both a high-level architecture (HLA) interface and a Windows component object model (COM) services interface. Using IMPRINT and these interfaces, the CART program successfully developed and integrated complex human performance models with a number of different system simulations, mission environments, 3-D human models and visual scenes, and data visualization environments (Doyal, Brett, Martin & Barbato, 2007). This demonstrated a major advancement, enabling HPMs to participate in larger simulation environments and even to interact with live players in those environments.

The ability to build higher-fidelity, interactive HPMs yields new opportunities for research and applications across a wide range of human-performance related domains. One such area is *naturalistic decision making*. Naturalistic decision making (NDM) can be defined simply as "the way people use their experience to make decisions in field settings" (Zsambok, 1997). Zsambok also offers a more detailed description of NDM:

> The study of NDM asks how experienced people, working as individuals or groups in dynamic, uncertain, and often fast-paced environments, identify and assess their situation, make decisions and take actions whose consequences are meaningful to them and to the larger organization in which they operate. (Zsambok, 1997, p 5).

By its nature, the study and/or training of NDM focuses on decision making as it actually occurs in the real-world environment, as opposed to decision making in laboratory tasks with clear "right" and "wrong" results. Recreating a more realistic and complex task environment to support NDM research and training is a significant undertaking. Often these environments are highly complex and dynamic and involve multiple systems and individuals with whom the decision maker may interact.

Use of multiple real-world systems and live personnel to serve as a "supporting cast" in decision making research and training is often cost-prohibitive. However, to the extent that we can model complex systems and individuals to create immersive environments through simulation, we can begin to study NDM in realistic and complex, yet controlled and affordable settings. The effort described here explores the approach and technology associated with creating such an immersive, simulation-based environment.

Concept of a Corrective Action Simulator

In late 2006, AFRL/RHCP initiated a 13-month effort to demonstrate how human performance modeling and integration approaches like those developed under the CART program might be applied to create an immersive environment that supports interaction between a live operator and synthetic teammates working to perform a complex task. The idea was to create a true immersive environment using human performance modeling technology, with a focus on providing realistic scenarios, cues and responses, which are critical to stimulating the decision making process (Cannon-Bowers & Bell, 1997).

The context chosen for this demonstration involved a live player in a supervisory role who would observe the actions of synthetic subordinates and intervene as necessary to initiate "corrective action" when the subordinate was observed to make an error of omission or commission. This concept of a *corrective action simulator*, or *CAS*, environment was thought to pose a significant human performance modeling challenge because of the many simulation components required to create an immersive environment for the live operator. Specifically, the player must observe human forms (dynamic 3-D models), interacting with systems (dynamic workstations), and talking amongst themselves to perform a given set of tasks. Further, he/she must be able to interrupt these synthetic teammates and offer verbal suggestions that the synthetic players can understand and carry out to correct previous errors.

Project Goals and Objectives

The goal of the current effort was to identify a methodology for creating a Corrective Action Simulator - or "CAS"- as described above and to explore its feasibility by creating a prototype simulation that could be used to demonstrate the concept. The focus of the effort was on determining an engineering solution to building such a device and assessing its feasibility. Though this involved a significant effort in the areas of software/hardware engineering (design, development, integration & test), it also included an up front focus on the human-centered design. That is, it also examined a means of determining the *content* for a CAS and how that content could help achieve training objectives.

To help guide the design and development of the CAS prototype demonstration system, a number of desired system characteristics were identified. These characteristics, described below, were treated as objectives during the course of the project.

Focus on supervisory role. One key objective for the CAS prototype demonstration was to support the training of an individual who serves in a supervisory capacity. AFRL/RHC has long been interested in various team interactions and the degree to which such interactions are influenced by a number of factors including technology, training, stress, and uncertainty. On this project, researchers were interested in presenting a situation where a supervisor is required to maintain a degree of situational awareness, observe a situation where inappropriate and/or insufficient actions are being taken by the subordinate teammates, and intervene in the situation to ensure that corrective action is taken. Further, in the given scenario of interest, the CAS prototype demonstration system was required to demand some degree of decision making on the part of the live individual (i.e., the supervisor).

Establish an immersive environment. A second objective for the CAS prototype demonstration system was to create a simulation-based immersive environment. In order to place a live player in a realistic supervisory environment, allowing him/her to recognize a potential problem, to interact with synthetic subordinates to take action, and to see the results of the corrective action; it was necessary to create an environment that supported multiple, realistic cues and also enabled realistic control mechanisms. Specifically, the objective was to support both visual stimuli (e.g., 3-D representations of subordinate teammates and their workstation displays showing the "state of the world") as well as auditory stimuli (voice interactions among team members) in an immersive environment that would also allow the supervisor a degree of verbal interaction with the synthetic subordinates.

Reflect a real-world event within an AWACS domain. The final major objective of the CAS prototype demonstration system was to convey a "real-world" situation in which a supervisor was required to interact with subordinates in an effort to effect a corrective action. To demonstrate the CAS concept in a realistic scenario, AFRL/RHC wanted to re-create a real-world situation in which an Airborne Warning and Control System (AWACS) Senior Director (SD) had to observe and correct a Weapons Director (WD) error in order to achieve mission success. Given the dynamic and complex nature of controlling air space and the roles and interactions among an SD and WDs in performing this function, the AWACS environment seemed an ideal one for demonstrating the CAS concept. Thus, the effort focused on the AWACS domain.

Approach to Developing the CAS Prototype Demonstration System

Below, we outline the process undertaken in the development of the CAS prototype demonstration system. These steps generally resemble the approach that would be taken to develop a system for an end user with specific training objectives and requirements. However, given that the end product of the current project was to be a conceptual prototype for demonstrating the technology, the focus of this effort was weighted less toward the training system requirements analysis and more toward the system engineering component.

Identifying Training Objectives

The first step in developing the system was to identify training objectives to be supported by the device. As stated above, a high-level objective was to work within an AWACS domain, focusing specifically on the SD as he/she interacted with WDs. In initial discussions with SD/WD subject matter experts (SMEs), it was determined that one difficult-to-learn skill for many SDs is the ability to monitor both AWACS situation displays and multi-channel audio to maintain situational awareness of both the environment and the WDs' interactions with players in that environment. Specifically, in high-tempo situations, it can be difficult to recognize the implications of various voice interactions among the WDs and the aircrews with whom they are interacting. Often there are key pieces of information in these voice interactions that are subtle, fleeting and available nowhere else. If there is an error or an omission in the communication, or if the implications of a communication are not recognized, the mission can be significantly impacted. The SMEs with whom we spoke felt that key training objectives should include teaching the SD to perceive and recognize the impact of rather subtle nuances in verbal interactions and also to convey the importance of using both that awareness of the situation and the SD's operational knowledge to identify the best course of action for a given situation.

Identifying an Incident of Interest

With these objectives in mind, we set out to identify candidate real world AWACS incidents in which the SD was required to intervene after observing an error of omission or commission made by a WD. To identify such an incident and to subsequently fully understand how it transpired, we employed a knowledge elicitation technique referred to as the Critical Decision Method (Klein, Calderwood, & MacGregor, 1989). Described by Klein and his colleagues as a "retrospective interview strategy that applies a set of cognitive probes to actual non-routine incidents that required expert judgment or decision making", the Critical Decision Method (CDM) seemed an ideal tool for capturing various incidents of interest and subsequently decomposing a given incident to such a level that it could be accurately recreated in an immersive simulation environment.

To identify candidate AWACS incidents, we conducted a series of individual interviews with four AWACS SD/WD subject matter experts with varying levels of AWACS experience. Table 1 outlines the background/experience of the four SMEs. Using the

CDM, we asked each SME to recall situations in which WDs made errors requiring the SD to become involved and to initiate some form of corrective action in order to ensure a successful mission outcome. We did not restrict these to "success stories" in which the error was detected and corrected, but rather we also allowed for those occasions when the mission was negatively impacted because an error was *not* detected or dealt with adequately. Each SME relayed to us at least one critical incident that fell within these criteria. Initially, SMEs were asked to provide only an overview of the incident. They generally set the scene, describing major events, decision points and outcomes. After reviewing the critical incidents described by each SME, AFRL/RHCP chose one particular incident upon which to build the CAS prototype demonstration system. This incident, as relayed by SME 4, involved an AWACS supporting dynamic targeting operations in a wartime environment.

Table 1. Subject Matter Experts' Relevant Experience

Subject Matter Expert	Relevant AWACS Experience
SME 1	2 yrs as WD, 1 yr as SD.
SME 2	10 yrs as WD, SD, and SD/WD instructor/evaluator. Also 8 yrs in a Control and Reporting Center (CRC)
SME 3	9 yrs in AWACS – 2 yrs as WD, 7 yrs as SD. 3 yrs in CRC.
SME 4	2 yrs as WD, 3 yrs as instructor WD, 3 yrs as SD, 8 yrs as Mission Crew Commander (MCC) and/or instructor/evaluator MCC.

Using the CDM, we conducted a more in-depth interview with SME 4 in which we obtained details about the dynamic targeting event he had described earlier. During this interview and subsequent discussions, we learned details about the exact timeline of events, the triggers associated with those events, key decisions points, decision criteria, and outcomes. Though we will not recount the details of that interview here, the overview provided in Table 2 should provide the reader with sufficient detail to understand the key aspects of the scenario and the corrective action that was required.

Table 2. Scenario Overview

Setting	AWACS was performing a combat mission in support of dynamic targeting. Key role was to match strike aircraft against dynamic ground targets.
Incident Overview	AWACS received call for strike against an armor column that was approaching a friendly troop position. AWACS WD paired a strike aircraft against the armor column and handed aircraft off to ground controller for final strike coordination. Strike aircraft was unable to complete the strike mission because it lacked the appropriate weapons mix to prosecute the armored target. Friendly troops were forced to withdraw.
Key Event 1	AWACS receives indication of a mobile surface to air missile threat in area of responsibility (AOR) (this served a distraction from the dynamic targeting activities)
Key Event 2	Strike aircraft "Sword" lead checks-in with AWACS Check-in WD as it enters the AOR. (Other available strike aircraft have already checked in at this point).
Key Event 3	AWACS receives notice of dynamic target in AOR
Key Event 4	AWACS receives request for pairing against the dynamic target
Key Event 5	AWACS Dynamic Target WD pairs "Sword" against dynamic target and hands Sword off to ground-based Joint Terminal Attack Controller (JTAC)
Key Event 6	AWACS receives word that Sword is not equipped to prosecute target
Key Event 7	AWACS receives word that friendly troops had to withdraw—mission against dynamic target failed
Critical Error 1	Sword lead fails to provide "status" data at check-in. Sword's state was different than what was represented in the Air Tasking Order (i.e., different than "fragged"). Check-in WD also fails to request Sword "status" at check-in. This results in insufficient situation awareness regarding the actual weapons load on the aircraft.
Critical Error 2	Dynamic Target WD, making invalid assumption regarding Sword's sensor/weapon state, pairs Sword against a target type that Sword is not configured to prosecute.
Relevant Cues	Sword's radio call at check in, check-in calls of additional strike aircraft in the AOR, ATO & SPINS data with fighter's sensor/weapon configurations, target description
Contributing Factors	High workload – many aircraft checking in, some going to tanker, dealing with emerging threat situation
What-if Scenario	If Check-in WD had requested Sword status and Dynamic Target WD had integrated this information with knowledge of target, a different (appropriately equipped) strike package could have been paired against the target in time to prosecute it, such that friendly forces would not have had to withdraw.
Corrective Action Opportunity 1	SD should recognize the lack of situation awareness and direct Check-in WD to query Sword's state
Corrective Action Opportunity 2	Once Sword's state is known, SD should ensure the Dynamic Target WD does not pair Sword against the armor column and instead pairs an appropriately configured strike asset.

Designing and Developing CAS Prototype Demonstration System

Scenario Development. The first step in the development process was to further elaborate details of the scenario of interest, identifying all of the key players and their roles in the scenario. This, coupled with training objectives, helped form the basis for developing system requirements. Working with the SME 4, whose critical event was chosen for implementation, the CAS developers identified the specific air and ground entities and key AWACS personnel involved in the scenario. These are listed in Table 3.

Table 3. Key Players in the Scenario

Key Entities – Air and Ground		**Key AWACS Personnel**	
"Bronco"	2 F-16CJs Role: Dynamic Targeting	"SD"	Senior Director (live player)
"Claw"	4 F-15Es Role: Dynamic Targeting	"WD2"	Check-In / Tanker Controller
"Sword"	2 F-15Es Role: Dynamic Targeting	"WD3"	Dynamic Targeting Controller
"Cylon"	4 F-15Cs Role: Defensive Counter Air	"ECO"	Electronic Combat Officer
"Exxon"	2 KC-135s	"MCC"	Mission Crew Commander
"Mojo"	AWACS Weapons Controllers	"Engineer"	Flight Engineer
"Sabre"	JSTARS Mission Crew		
"Tiger 13"	SOF Team JTAC		
Track J5033"	Enemy Armor Column (the dynamic target)		

Although the key error in the scenario centers on the check-in and tasking of Sword flight, a number of other aircraft are present in the scenario and under control of the AWACS. To support the dynamic targeting mission, three flights of strike aircraft are available. In addition to Sword, strike assets include another flight of F-15Es ("Claw") and a flight of F-16s ("Bronco"). Other fighters in the scenario include a flight of F-15Cs ("Cylon") who serve a defensive counter air (DCA) function. In this scenario, the DCA fighters are present but do not play a critical role. In addition to the fighters, there are two KC-135 tanker aircraft ("Exxon 32" and "Exxon 33"), a Joint Surveillance Target Attack Radar System – JSTARS aircraft ("Sabre") and the AWACS aircraft whose controllers use the callsign "Mojo".

Orbits for the various aircraft and positions of ground entities were then identified. For the purposes of keeping the demonstration system unclassified and also to use an area with which any USAF SD would be familiar, the location for the scenario was placed at Nellis AFB, Nevada. Figure 1 illustrates the relative position of key entities in the scenario. On the eastern side of the region are friendly aircraft orbits. These include two tanker orbits, the Joint STARS orbit, and the AWACS orbit. In addition, there is a marshaling area, where the fighters go after they arrive at the check-in point, and an area where the DCA fighters are on combat air patrol. To the west lies the target area, composed of three "kill boxes". Each kill box is subdivided into nine "keypad" locations, numbered one to nine and corresponding to the relative position of numbers on a telephone keypad. The dynamic target in the scenario, a group of enemy tanks and armored personnel carriers (APCs) moving toward a friendly position, appears in Kill Box 81 Juliet, Keypad 6.

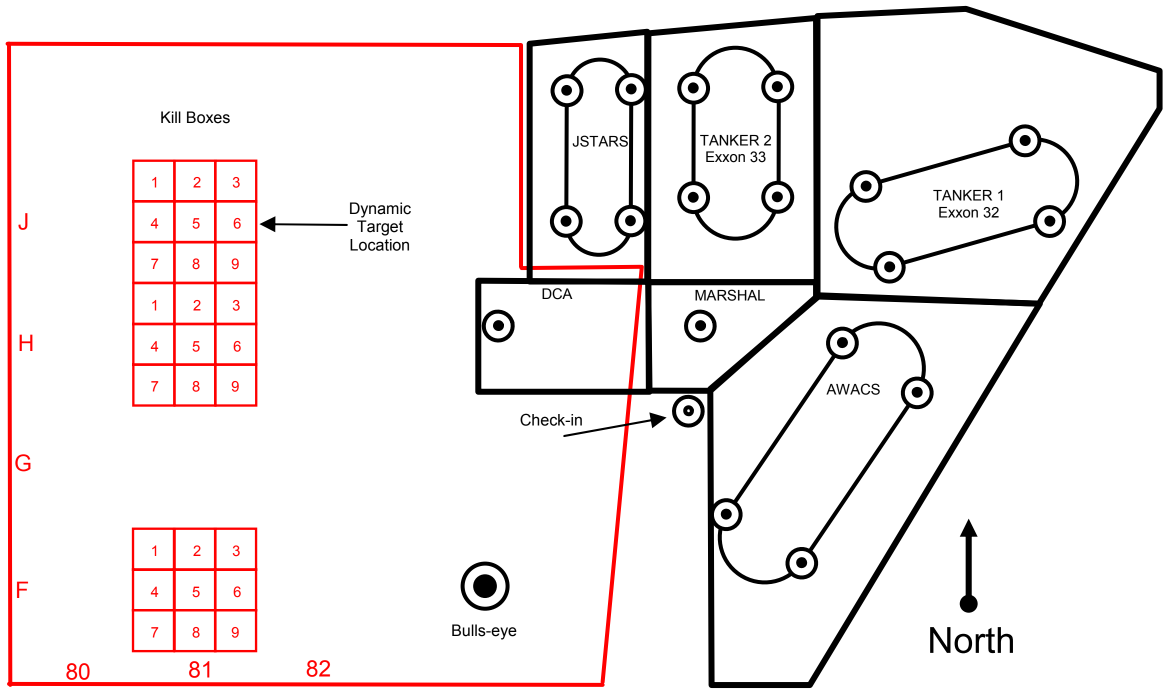


Figure 1. Relative position of entities in the CAS prototype demonstration scenario

Requirements Development. Based on an understanding of the incident to be simulated, the entities involved and their roles, and the training objectives to be accomplished; functional system requirements for the CAS prototype demonstration system were developed. A complete listing of the system-level requirements is provided in Appendix 1. It was determined that the system would be comprised of four primary subsystems: (1) a multifunction software component that would represent the entity tracks, control interactions among subsystems, and also support the system's graphical user interface (GUI); (2) a software component that would control the appearance of virtual WD displays, (3) a software component that would represent the appearance, behaviors and decisions of the synthetic players; and (4) a speech recognition/synthesis system.

The requirements development effort also involved identifying a training strategy that defines how the simulator will be used as a training device. Desired attributes of the system included (1) an automated introduction in the form of a briefing, (2) a "freeplay" mode that would allow the simulation to run without interruption or feedback, and (3) a computer-based training or "CBT" mode that would monitor trainee actions and intercede with cues if the live SD does not take corrective action within a given duration after a WD error. Within the systems requirements document shown in Appendix 1, these training component requirements are rolled under the "Track Generator Middleware" heading. This is because the Track Generator Middleware software component also served as an executive that controlled the CBT-related functionality.

Upon completion of the system-level requirements specification, in which functional requirements were assigned to each of the four subsystems, project engineers decomposed functional requirements for each of the subsystems into detailed software requirements (not described here). Each software requirement was derived from a higher-level system requirement such that bi-directional requirements traceability was maintained. Below, we provide an overview of the function and development of each of these subsystems. As a final step in requirements development, a system acceptance test plan was developed. This test plan identified test procedures for identifying whether each system-level requirement was implemented.

AWACS Corrective Action Simulator Tools Software. A software component we refer to as "ACAS Tools" was developed to serve a wide variety of functions within the CAS prototype demonstration environment. Developed using Microsoft Visual C++, ACAS Tools is primarily a reuse component from AFRL/RHC's CART program that provides two key functions: (1) It generates and publishes the air and ground tracks represented in the CAS prototype demonstration system; and (2) it acts as the focal point for all communications among various components of the system.

The track generator function produces a list of entities that are to be made visible on each virtual WD display. It creates the entity representation (e.g., tank, aircraft.) and controls the path each entity takes. It also periodically captures data regarding each entity in the simulation (e.g., type, position, heading, velocity), and makes these data available for use by other components in the simulation. Under the CART project, this ability to capture and transmit entity state data was limited to ground entities. With the CAS project, an ability to capture and transmit airborne entity data was added. The modeled entities include various fighter aircraft, tanker aircraft, airborne ISR assets, a static JTAC position, a surface to air missile (SAM) threat, and a convoy of tanks and armored personnel carriers. The JTAC and SAM entities are stationary during the scenario, whereas all other entities are moving. As the scenario progresses, the track generator function determines and makes available the status of each entity in the simulation, representing the list of tracks that would be seen by the AWACS radar, fed to the AWACS from another sensor platform, and/or entered by an operator. This list of entities is sent to both the weapons director display content processor (WDDCP) software and the human performance model, both discussed later in this section. ACAS Tools communicates with the WDDCP via data files on shared drives and also publishes entity state data via a Distributed Interactive Simulation (DIS) interface. It communicates with the human performance model via HLA 'Data' interactions or messages using the Defense Modeling and Simulation Office (DMSO) Run Time Infrastructure 1.3NG-Version 6. The ACAS Tools and human performance model components are also synchronized via the HLA time management facilities.

ACAS Tools not only produces track data for entities, it also generates voice communication content for the fighter aircraft entities it represents. In the CART version of the software, aircraft were modeled with a simple flight model and responded to tasking requests. Most tasking requests involved flying to a new location or, in the case of attack, dropping a weapon on a specific target. For the CAS project, a capability to generate appropriate voice messages was added. These messages include a check-in message generated as an aircraft reaches the check-in location, responses to a WD's check-in messages (e.g., answering a 'check state' request), and other time-based messages (e.g., checking in with WD at later time or checking in with Dynamic Targeting WD on a different frequency). Once ACAS Tools determines the appropriate voice message content to be sent from one of the aircraft, it passes this message to the speech synthesis system via a socket interface. The speech synthesis system, described below, then plays the appropriate audio file for the given message.

Several additional ACAS Tools capabilities were developed to support the CAS prototype demonstration. ACAS Tools includes a Simulation Control Panel represented with a graphical user interface. This allows a live user to control starting, stopping, restarting and mode setting of the simulation. It also supports functionality to give the demonstration a degree of CBT characteristics. These characteristics include an introductory presentation that provides user instructions and scenario information, pop-up “hints” that are provided to the user when action is needed, a data collection system for tracking the timing of key events and actions taken by the live player, a simple scoring mechanism for evaluating the user’s performance, and a de-brief presentation that gives the user feedback regarding his/her performance. ACAS tools also serves as a repository for data representing initial settings and synthetic operator control inputs to the virtual weapons director displays, storing and passing display setting data including map centers, map scale, track tagging information, and free text message content. Finally, ACAS Tools receives inputs from the human performance model regarding desired physical movements of the synthetic WD’s (e.g., body position and actions) and passes these to the 3-D visual environment-- Boston Dynamic’s DI-Guy™ Software Development Kit (SDK). The details associated with all interactions between ACAS Tools and other system components were specified in the CAS project’s interface control document, which served as a key reference during design and development of the system.

Virtual Weapons Director Displays. A key component in creating the desired immersive environment for the CAS prototype demonstration system was a representation of the WDs’ workstations. Although the chosen scenario and the WD errors that it entails revolve primarily around voice interactions and not control/display manipulation, we felt it important to provide a limited representation of the WD console. This would allow the SD to observe the WDs' high-level interactions with their workstations and also to perceive the state of the airspace, including the position of the strike aircraft relative to the dynamic target.

To support this capability, a CAS component referred to as the "Weapons Director Display Content Processor" (WDDCP) was developed. This system component, developed in Java 1.5 and utilizing the OpenMap™ toolkit, provides a very limited emulation of an AWACS situational display console. A screenshot from the WDDCP display presentation is shown in Figure 2. The display includes an overlay indicating named area perimeters. This overlay is generated via a data file and is easily configured. The display also presents track symbology indicating a track's type, geographic position, velocity vector, track history, and a text field indicating callsign, identification-friend-or-foe (IFF) data, and altitude (i.e., "tag" data). In addition, a "bulls-eye" (indicated by a '+' symbol) represents a point known only to the friendly forces. This point is used as a relative reference in range and bearing calls such that actual coordinates do not have to be used. In addition to the display characteristics, the WDDCP supports a limited degree of display manipulation. A set of scale expansion buttons enable zooming in on the display up to a factor of 32x. A cursor offset capability enables the view to be centered on a user specified geographical point and then reset to the original center point if desired. The WD display also includes a mission clock that displays mission time in a HH:MM:SS Zulu time format.

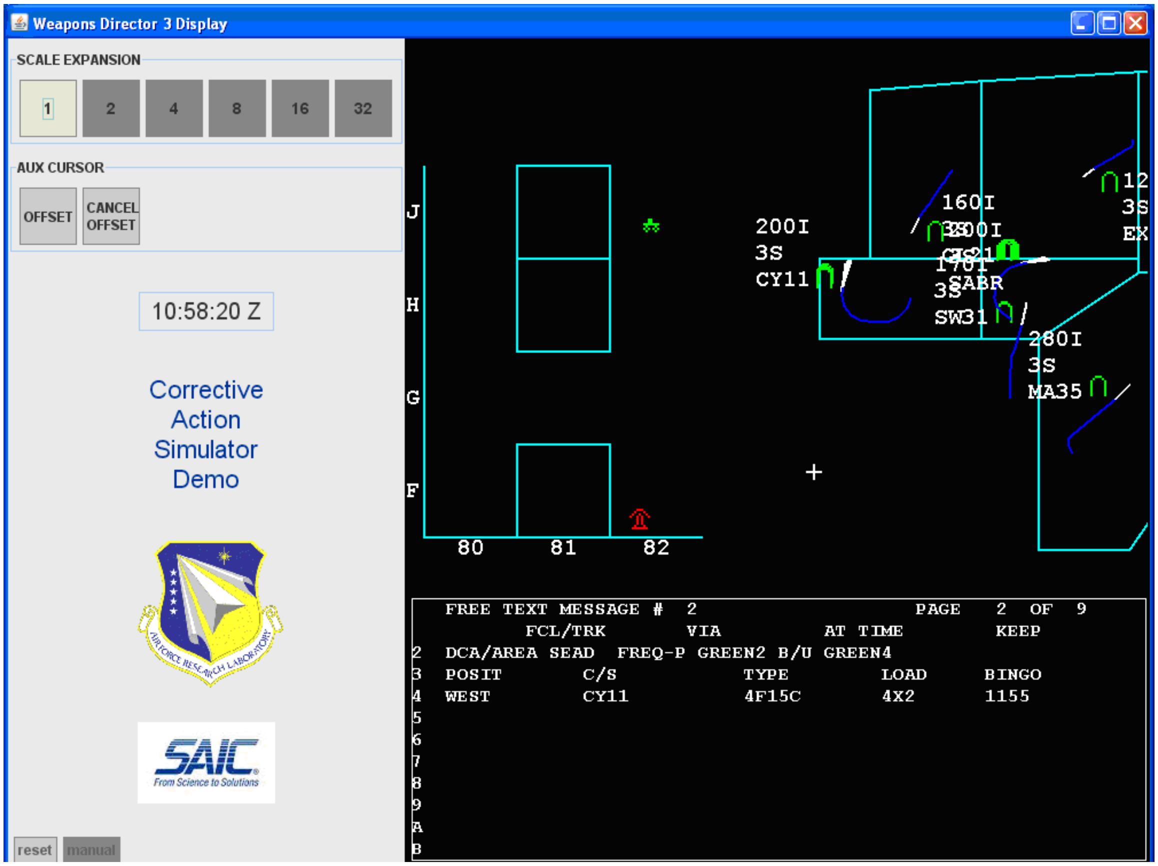


Figure 2. Weapon Director Display

This WD display console capability was designed to be controlled by the synthetic WDs in the simulation. However, it also includes a “manual” mode that allows direct manipulation by a live operator during runtime, overriding any inputs from the synthetic WDs. This manual override was included as an engineering test and demonstration mechanism and would not typically be used during a true training session.

One final feature of the WDDCP is that it exports snapshots of the display every twelve seconds. These bitmap files can then be accessed by the 3-D visualization tool that depicts the synthetic characters in the AWACS environment. This allows a live player to see the current WDDCP screens depicted within the 3-D visualization environment.

Synthetic Weapons Directors. Another key component of the CAS prototype demonstration is the representation of WDs with whom the live player interacts. This includes both a behavioral representation (decisions and actions taken by the synthetic WDs) and a 3-D visual representation. The behavioral representation is instantiated in a human performance model created using the IMPRINT modeling environment. The IMPRINT model includes a task-network-based representation of two weapon directors (a Check-in WD and a Dynamic Targeting WD) as well as other AWACS personnel required by the scenario (a Mission Crew Commander, an Electronic Combat Officer, and a Flight Engineer). For simplicity, the JTAC contact (not an AWACS crew position) was also represented in the IMPRINT model in order to facilitate generation and passing of needed stimuli. Relative to these other personnel, the WD's are represented at a higher level of fidelity, as they are the primary focus of the scenario.

All input and output data (primarily voice messages from personnel represented in the model) are created using HLA 'data' interactions. The incoming data provide the stimulus for the WDs to act upon. It represents the information the synthetic WDs hear on their headsets (intercom or radio) and see on their displays. The speech input enumerations include voice messages from onboard personnel as well as the live Senior Director and communications modeled as coming from the external aircraft. Other incoming data includes track lists, asset positions (KC-135s, AWACS, etc), ground entity positions, and any time-based simulation injects (e.g., JSTARS reporting tracks, the Engineer reporting a new aircraft bingo time, a threat being active.) Upon receiving this input, the synthetic players make decisions and then take action, usually resulting in outputs composed of voice interactions or interactions with the virtual weapons director display.

The outgoing voice messages (i.e., phrases to be spoken by the synthetic WDs) represent the WD responses to fighter aircraft, to each other, or to the live Senior Director. These data also include any of the appropriate metadata (e.g., the required bearing, range, & altitude) associated with a given type of verbal interaction. The HPM also sends out the commands for the appropriate avatar / character movements. These include movements such as typing or moving a mouse or trackball when working a check-in, turning to another crewmember to talk, and even the simple blinking of eyes. The aircraft re-tasking commands are administrative functions that provide the stimulus for the ACAS Tools-generated aircraft to fly to new commanded destinations. For example, when Bronco lead checks in and is directed to the tanker, the appropriate re-task commands are sent to the two aircraft that compose the Bronco flight to direct them to the tanker location.

The other component of the synthetic WD representation is the 3-D visual representation that a live player can observe. Development and rendering of the WDs and the AWACS interior is implemented using Boston Dynamic's DI-Guy environment. Figure 3 depicts the live player's view of the WDs at their consoles. The physical AWACS representation, an OpenFlight model imported into DI-Guy, was custom-developed for the CAS demonstration. It shows the fuselage interior as well as the WD consoles and chairs. In addition, a "window" is overlaid on the display area of the consoles. This window is capable of receiving texture updates from ACAS Tools at twelve-second intervals, allowing the virtual weapon director display content for each WD to be represented and updated within the 3-D visual environment.

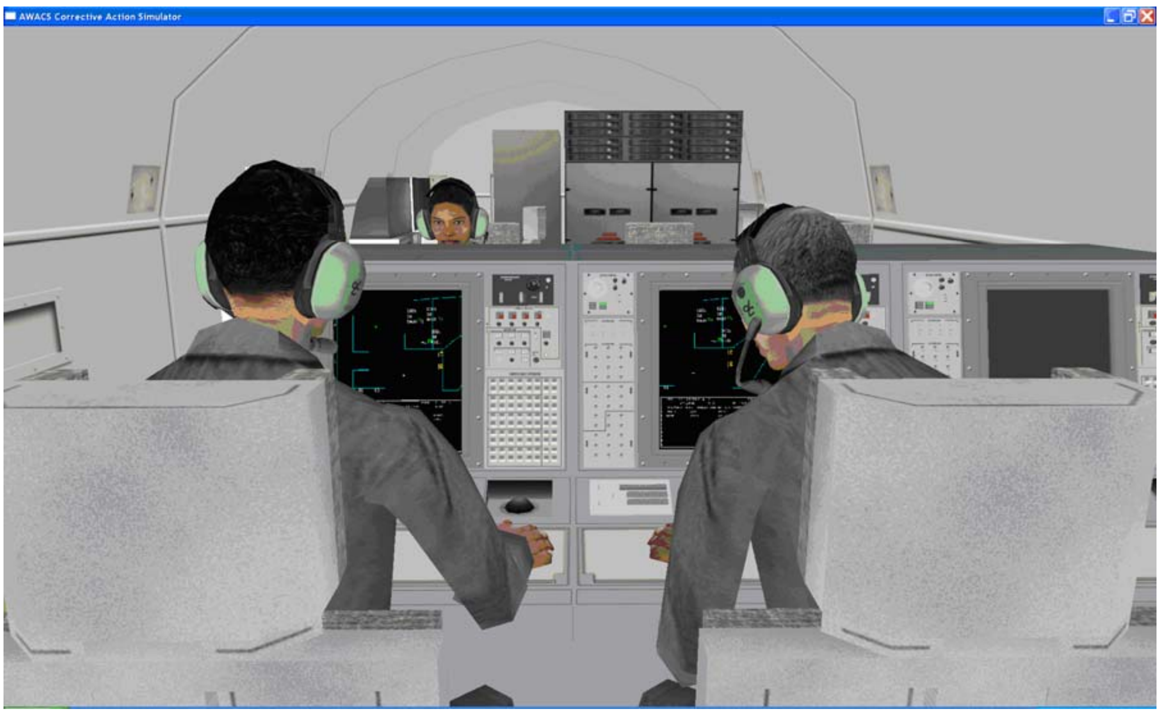


Figure 3. CAS's 3-D representation of SDs in AWACS

The WDs in the visual environment are represented using characters from the DI-Guy character set and also DI-Guy's "expressive faces" capability, which allows the lip movement of DI-Guy characters to be synchronized with the audio files that play character's spoken phrases. Their observable movements (i.e., blinking, turning or orienting, typing or using a mouse/trackball at the console) are all built-in actions available within DI-Guy. Commands for manipulating the character actions are generated by the human performance model and sent to ACAS Tools, which in turn, passes them to the DI-Guy SDK for rendering.

Speech Recognition and Synthesis. To help create the immersive environment for the live SD, we implemented a speech recognition and synthesis system that allows the SD to hear voice interactions among the AWACS crew members and between the WDs and pilots of the aircraft under AWACS control. In addition, this system is designed to recognize a set of phrases made by a live operator, allowing him/her to verbally interact with the synthetic WDs.

To identify both the radio & intercom calls that the SD would hear during the scenario and the potential phrases that the SD might initiate in response to an observed WD error, we worked with AWACS SMEs to "script" the scenario. SME 4 was tasked with scripting the radio and intercom interactions for the event. The script included phrases spoken by all key participants in the scenario including the WDs, other AWACS personnel heard on the intercom, and the pilots of friendly fighters who were checking in on the radio. This "baseline" script reflected the voice interactions that would occur during the scenario assuming no intervention by a live SD. Upon completion, the script was reviewed and edited by two current AWACS SDs at Tinker AFB, OK.

In addition to the baseline script, we worked with AWACS SMEs to identify potential voice interactions that would be initiated by a live SD in the context of the given scenario and the replies that the WDs would make in response to the SD. Specifically, we focused on understanding the desired actions an SD would want to take or information he/she would seek, as well as the phrasing he/she would use to request/direct this of a WD. Referring back to Table 2, note that the critical errors to be performed by the WDs and the corresponding potential corrective actions to be made by the SD involve the lack of state data from Sword flight at check-in and the subsequent tasking of Sword against the column of tanks and APCs. Potential SD-initiated communications associated with these errors would include a directive to the WD to check the state of the strike asset and also a directive to recall or "reset" Sword and to retask a different strike asset against the target. To support these interactions, developers worked with SME's to form a grammar set for the speech recognition system. SME's identified various phrases and keywords that SD might use to communicate these intentions. The grammar sets for the speech recognition system are listed in Table 4.

In the CAS demonstration, speech recognition is accomplished via a web-enabled speech application that incorporates the Microsoft speech recognition engine. Figure 4 illustrates the live operator interface to the speech recognition application. (Note: the interface in the figure is set to show optional active debug information that can be suppressed). The operator selects the "Push to Talk" button either via a mouse or a special foot-switch apparatus. The live player speaks the command and the speech recognition system compares the utterance to predefined grammar templates that identify valid phrases and variants of phrases to be recognized. If recognition is indicated by the speech recognition engine, key sub-phrases are identified in an utterance (e.g., "check state", "say again") to

Table 4. CAS Speech Recognition in Grammar Sets

Function	Grammar Recognized	Examples
Repeat Last Phrase	**WD**[*2 or 3*], SD: **Repeat** that **WD**[*2 or 3*], SD: **say that again** **WD**[*2 or 3*], SD: **say it again** **WD**[*2 or 3*], SD: **say again**	WD2, SD, Say again
Repeat Callsign	**WD**[*2 or 3*], SD: **say again callsign**	WD3,SD: say again callsign
Retask	**WD3,**SD: **Retask** [*Bronco or Sword or Claw or Cylon*] to [*81 Hotel or 81 Foxtrot or 81 Juliet or Armor or Tanks*]	WD3,SD: Retask Bronco to 81 Foxtrot WD3,SD: Retask Claw to tanks
Untask Aircraft	**WD3**,SD: **Skip it** [*Bronco or Sword or Claw or Cylon*] **WD3**,SD: **Reset** [*Bronco or Sword or Claw or Cylon*]	WD3,SD: Skip it Sword WD3,SD: Reset Cylon
Request Aircraft State	**WD2**,SD: check **state** [*Bronco or Sword or Claw or Cylon*] **WD2**,SD: what is the **status** of [*Bronco or Sword or Claw or Cylon*] **WD2**, SD: what **state** [*Bronco or Sword or Claw or Cylon*]	WD2,SD: check state Bronco WD2,SD: status Sword WD2,SD: what state Cylon
Request Playtime for Aircraft	**WD2**, SD: check **playtime** for [*Bronco or Sword or Claw or Cylon*]	WD2,SD: check playtime for Claw
Request Weapons Load	**WD2**,SD: check **weapons load** for [*Bronco or Sword or Claw or Cylon*]	WD2,SD: Weapons load Cylon
Request Check-in Status Info	**WD2**,SD: has [*Bronco or Sword or Claw or Cylon*] **checked** in?	WD2,SD: has Sword checked in
Acknowledge	**WD2,SD: copies checkin** **WD2,SD: copies [***Bronco or Sword or Claw or Cylon***] checkin** **WD2,SD: copies state** **WD2,SD: copies[***Bronco or Sword or Claw or Cylon***] state** **WD2,SD: copies**	WD2,SD copies Bronco checkin WD2,SD copies state WD2,SD copies
Roll Call	**Weapons**,SD: **roll call**	Weapons, SD: roll call
Comm Check	**WD**[*2 or 3*],SD:**COM Check** **WD**[*2 or 3*],SD: how do you **hear**?	WD2,SD: COM Check WD3,SD: how do you hear?

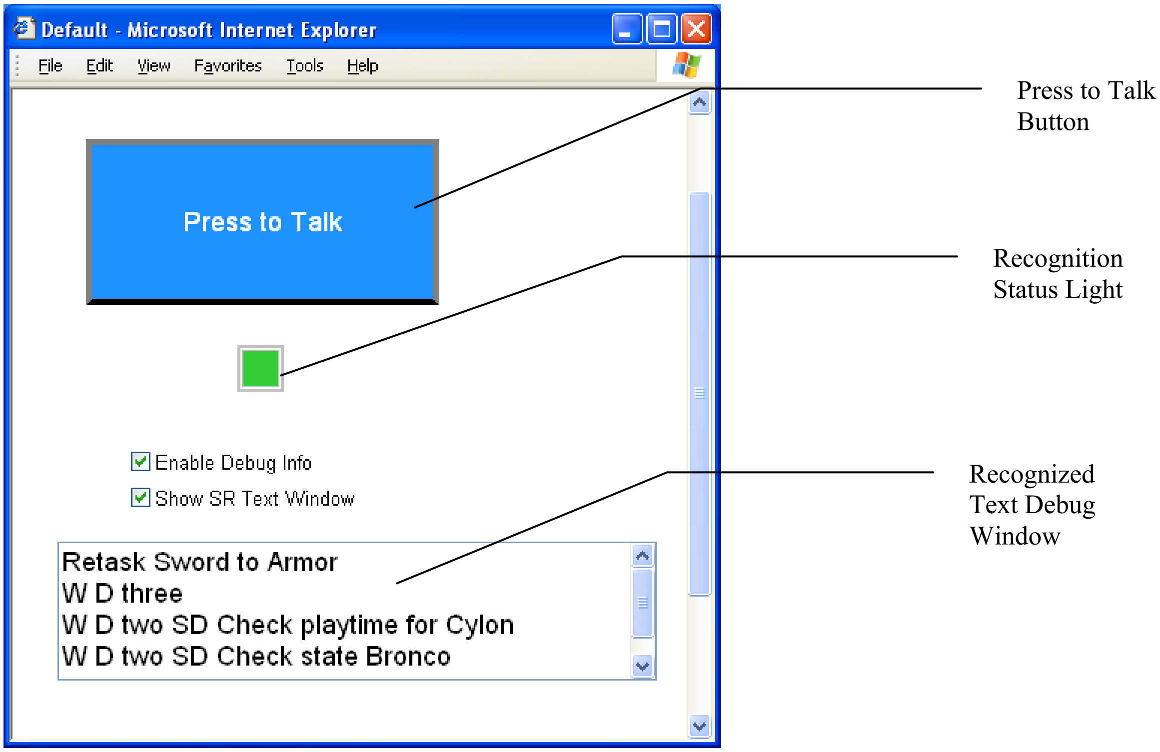


Figure 4. User interface for speech recognition system

determine the intent of command. The grammars are also used to identify "dynamic data" within a spoken utterance and these data are extracted (e.g., callsigns, target identifiers). If an utterance is only partially recognized, this dynamic information stored for transmission and data fields without recognized data are left set to a "not recognized" value. The HPM can subsequently use the "partial recognition" status as a trigger to perform a follow-on request for the missing information via a spoken prompt from one of the virtual characters. The ability of the software to recognize an utterance is communicated to the live player by setting the "Recognition Status Light" to green (indicating complete recognition), yellow (indicating partial recognition), or red (indicating no recognition). All of the recognized data from an utterance are encoded and sent as a message to the HPM via a socket interface.

A tool called *3D MP3 Sound Recorder G* ("free" version) was used to record the phrases and words for each speaker. To simulate the sound of these phrases being spoken over an intercom or radio, another freeware tool called *Soliton II* was used to degrade the sampled audio as appropriate. To create the effect of speech over the intercom, high and low-pass filtering of 300 Hz and 3 KHz, respectively, was applied. Simulating the quality of radio communication required further degrading of the audio files. In addition to the high and low-pass filters, the audio files representing those synthetic players speaking over radio had both noise and linear audio distortion applied. According to an AWACS SME who later viewed the system, the resulting audio quality adequately produced the effect of intercom and radio communications.

To generate the synthetic speech during runtime, the human performance model communicates with the speech synthesis task via a socket interface and indicates the speaker (voice), the phrase to be spoken, the selected voice degradation (intercom or radio), and any dynamic fields required by the phrase (e.g., distances, times, callsigns). The speech synthesis system locates the set of sampled voice files required to speak the phrase and splices these together into a single audio (*.wav*) file. Next, the speech synthesis system processes the audio to identify the phoneme and viseme information (basic units of speech in the acoustic and visual domains, respectively) that needs to accompany the audio file in order create lip-synced motion for the DI-Guy characters. This is accomplished by sending the *.wav* audio file to Haptek's *HapOGGFactory* tool, which adds the lip sync metadata and creates *.ogg* files. Finally, this file is copied to the laptop running the "ACAS Tools" software and a message is sent by the speech synthesis system indicating the completion of speech synthesis and the name of the *.ogg* file, at which point it is played by the system and heard by the live player.

System Integration and Testing. The CAS prototype demonstration system is designed to run on a set of four laptop personal computers (PCs) networked with a wireless router. The configuration is shown in Figure 5 below. Two laptops (Laptops "1" and "2" in the figure) are used to show the weapons director displays; one each for the Check-In WD and the Dynamic Target WD. Laptop 1 also hosts the human performance model that controls the synthetic characters' actions and decisions. A third laptop hosts the speech recognition and speech synthesis software components as well as a graphical user interface for the speech synthesis system. The fourth laptop hosts the remainder of the CAS software components including the 3-D virtual environment the track generator software, the simulation "middleware", a graphical user interface that includes pre-mission and post-mission briefing slides, and the data collection software. The systems are networked through a wireless router to achieve a cleaner hardware configuration.

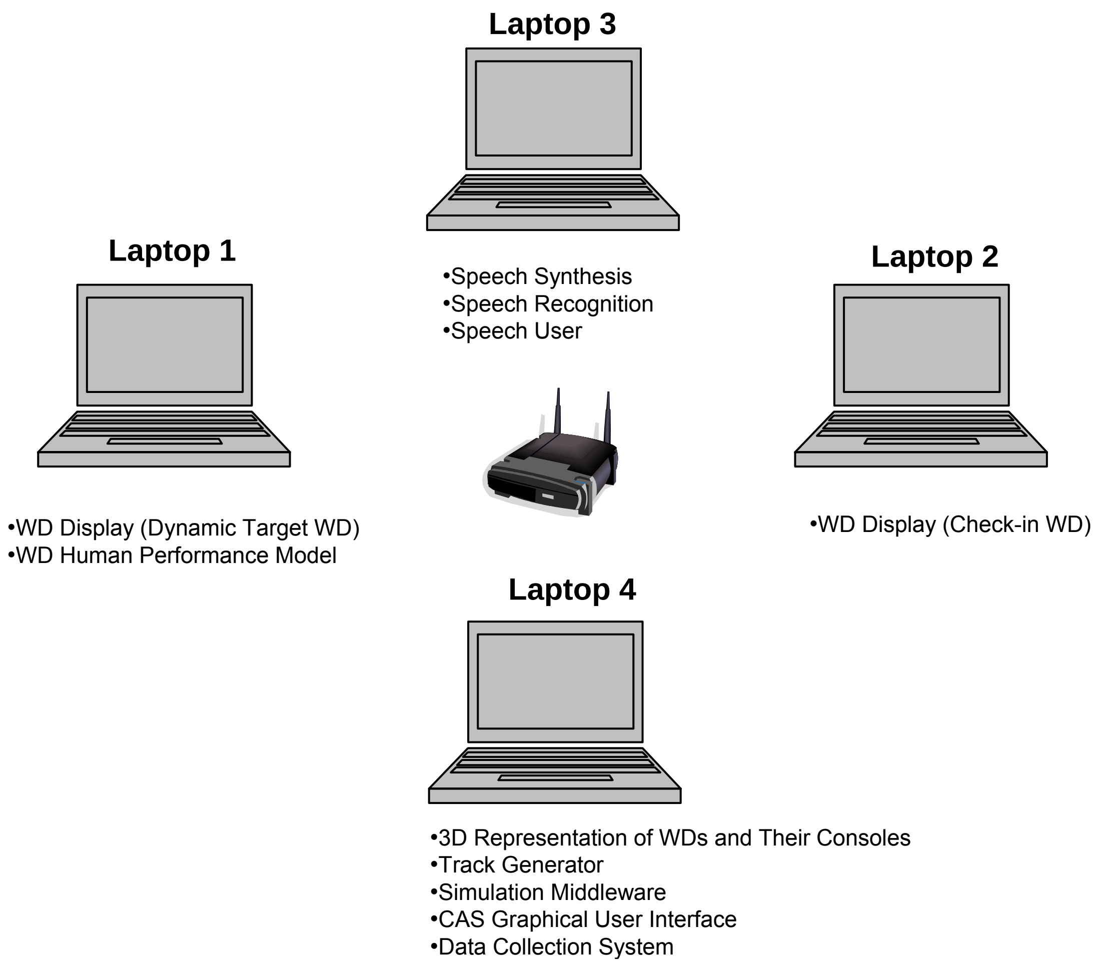


Figure 5. CAS prototype demonstration hardware & software configuration

Once the system components had been developed and tested individually, the system was integrated and tested. Much of this testing was informal, iterative, and involved ensuring that the appropriate entity states were being displayed on the WD Displays, that the speech synthesis system was generating the appropriate calls from the various players at the appropriate times and that the speech recognition system was understanding the defined live player inputs. Once the system components were shown to be behaving as intended, the system was demonstrated to an experienced AWACS WD/SD in an effort to solicit additional feedback and recommendations for improvement. Based on this feedback, a number of minor adjustments were made to the pre-mission briefing and to the grammars associated with anticipated SD speech inputs. The final set of testing involved formal execution of an acceptance test plan in which the project's software manager verified that the system performed in full accordance with the System Requirements Specification.

A Walkthrough of the CAS Prototype Demonstration

Before concluding the report, we would like to give the reader a feel for how a live player, acting as the SD, might interact with the CAS demo. Below, we step through a sequence of events that illustrate this interaction.

At the start of the demonstration, the live player is presented with a graphical user interface on Laptop 4 from which he/she can initiate a mission pre-brief. Once selected, a PowerPoint-based mission pre-brief presentation appears on the same laptop. The briefing slides are included in Appendix 2. The presentation briefly introduces the purpose and intent of the ACAS prototype demonstration system and then begins to set the stage for the AWACS scenario represented in the demonstration. This information includes an overview of the mission type, time and location; as well as strike asset data including a map of orbit positions, and the air tasking order (ATO) and special instructions (SPINS) that outline, among other things, the planned weapon load for each asset. The purpose of this briefing is to give the live player enough context and situation awareness to enable him/her to later recognize WD performance issues in the scenario and to initiate corrective action. After reading through the briefing slides at his/her own pace, the live player can close the pre-mission briefing and press a button on the GUI to start the scenario.

As the scenario starts, the 3-D environment appears on Laptop 4. On this display, the live player can see the two synthetic weapons directors sitting and working at their consoles. On Laptop 3, he/she sees the GUI for the speech system, including a Press-to-talk button that can be used when addressing the synthetic WDs. On Laptops 1 and 2, he/she sees the situation displays for the Dynamic Targeting and Check-in WDs, respectively. Through speakers attached to Laptop 4, he/she can begin to hear radio and intercom traffic coming from synthetic players.

Early in the scenario, the strike aircraft begin to check in. Bronco flight checks in first, followed by Claw. As Bronco checks in, the synthetic Check-in WD directs him to one of the tankers (Exxon 32) for refueling. Claw proceeds to the marshaling area. Soon after, the dynamic target appears in the scenario. Initially this appearance is noted by an alert of a JSTARS track. This alert comes to the WDs from the Electronic Combat Officer (ECO) over the AWACS intercom. Later in the scenario the WDs hear over the intercom that Special Forces on the ground have identified a column of tanks and APCs at the same location as the JSTARS track, indicating a positive identification on the track and confirming that the track is hostile. During the same few minutes that the target is emerging, the AWACS learns of an SA-8 surface to air missile site that has become active in the area. This communication is heard over the AWACS intercom and the Check-in WD then begins alerting the strike assets of the new surface-to-air threat.

During the time that both the threat and the target are emerging, Sword flight checks in. When checking in, Sword lead fails to provide status (including weapons load). Per the scenario, Sword's actual weapon load is different than what was originally planned. Although the ATO and SPINS, which the live SD sees during the pre-mission brief, indicate that Sword is carrying munitions suitable for use against armor, Sword's actual weapon load is primarily for use in air-to-air engagements and is ill-suited for armored targets[2]. Also in accordance with the scenario, the Check-in WD fails to query Sword lead on weapon state. This results in the Check-in WD having poor situation awareness regarding the strike platforms. If the live SD perceives this problem, he/she can take action to correct it. The corrective action would involve instructing the Check-in WD to check Sword's state. To accomplish this, the live player would press a footswitch to activate the speech recognition system and then issue a verbal command to the synthetic Check-in WD. If instructed to do so, the synthetic Check-in WD will verbally query Sword lead, who will then report his actual state data, which the live SD will hear.

Soon after checking Sword in, the Check-in WD hands Sword off to the Dynamic Target WD who, in turn, moves to task Sword against the target. Regardless of whether the Check-in WD requests state data for Sword, the Dynamic Target WD will assign Sword to the emerging armor target. This creates a second error in the scenario. This error,

[2] It is not uncommon for aircraft to be carrying munitions that differ from what was planned. In this case, it is standard procedure for the flight lead to pass this updated "state data" to the WD at check in.

theoretically, could have two causes: (1) poor situation awareness (SA) on strike aircraft weapons load results in an inappropriate weapon-target pairing (i.e., the manifestation of the first error), or (2) a failure to *apply* knowledge of the actual weapon load and target characteristics to achieve and appropriate weapon-target pairing. To implement the tasking, the Check-in WD instructs Sword to contact the JTAC for instructions on prosecuting the target. If no further corrective action is taken by the SD, Sword flight moves toward Killbox 81 Juliet and contacts the JTAC. After contacting the JTAC, Sword lead learns that they will be unable to complete the mission, and subsequently reports this information back to the Dynamic Target WD, at which point the scenario is ended. However, if the SD realizes Sword is not an appropriate choice for pairing against the tanks and APCs, he/she can verbally instruct the synthetic Dynamic Target WD to recall Sword and send Claw against the targets. If he/she does this, Claw heads toward Kill Box 81 Juliet, and contacts the JTAC, ending the demo scenario.

If the demo is run with the CBT mode turned on, certain hints will be presented to the live SD as the scenario unfolds. Specifically, 50 seconds after Sword checks in, if the live SD has not instructed the WD to check Sword's state, the simulation is paused and a pop-up briefing slide appears instructing the SD to request Sword's state data from the WD. After reading the slide, the SD can close it and press a button to resume the simulation. Similarly, 50 seconds after Sword is tasked to the target, if the live SD has not instructed the WD to reset Sword, a pop-up briefing slide appears instructing the SD to reset Sword and to send a more appropriate strike asset. This hint can also be acknowledged, which closes the slide and resumes the simulation. At the end of the demonstration, a final briefing slide appears. This slide shows results of the demonstration run. The key errors requiring corrective action are described and checkboxes indicate whether or not the live SD corrected the errors. In total, this demonstration scenario runs approximately 20 minutes.

Follow-on Avatar Development Effort

After the ACAS Demonstration technical effort had been completed and near the conclusion of the ACAS contract, AFRL/RHC was approached by a representative of the Air Education and Training Command (AETC) who had seen the ACAS prototype and was interested in the degree to which avatars could be rapidly developed, customized to resemble specific individuals, and made to support realistic looking gestures and movements that mapped to the content and emotion of the avatar's speech. To quickly address this question and provide representative examples of the products that could be developed, SAIC conducted a rapid avatar prototyping effort with the help of two avatar vendors: Boston Dynamics, Inc and Forterra Systems, Inc.

The avatar developers were provided with multiple high-resolution photographs of two Air Force officers in uniform. These photographs, provided by AETC, were shot against a green background from multiple angles to show each officer from all sides, including a top down view. In addition, AETC also provided two audio files containing short (1-2 minute) informal speeches delivered by the officers. The guidance given to the avatar developers was intentionally loose, allowing them "creative license" in developing their sample avatars. They were asked simply to create avatars that resemble the actual officers as closely as possible, to place the avatars in a scene of their choosing (e.g., at a podium, behind a desk etc.), to synchronize lip movement with the audio files, and to animate the avatars with realistic facial expressions and gestures that match what would be expected given the spoken content and emotion in the audio files. The vendors were given only seven days to create the prototypes and could choose whether to focus their time/effort on creating only one avatar (representing only one of the two officers) or to create avatars of both officers. In the end, the developers took different approaches. Forterra Systems chose to spread their effort over the creation of both officer avatars, whereas Boston Dynamics chose to focus their efforts on only one. Screenshots from the two Forterra-developed prototypes and the Boston Dynamics-developed prototype are shown below in Figures 6, 7, and 8.

Figure 6. Screen capture from Forterra Systems' avatar development effort (1 of 2)

Figure 7. Screen capture from Forterra Systems' avatar development effort (2 of 2)

Figure 8. Screen capture from Boston Dynamics' avatar development effort

The end product of this effort consisted of three video files, one for each of the avatar development efforts. Each video showed an officer walking onto a stage and delivering a short speech. During the speech all avatars exhibited physical gestures consistent with the spoken content and lips were synchronized with speech. Due to the viewing ranges, they did not seem to exhibit much range in facial expression; however, in working with the vendors, it was clear that this capability exists in the tools they used to create the avatars. After these avatar videos were delivered to AFRL/RHC, they were also passed on to AETC. The avatar videos were subsequently rolled into public AETC presentations describing how avatar technology might be used to support the future of Air Force Education and Training.

Conclusion

Lessons Learned

Developing the CAS prototype demonstration system proved challenging for the engineering team and provided a good learning experience for all involved. Below, we outline some of the key lessons learned along the way.

Interview Technique. As mentioned earlier, the technique we employed for capturing real-world events from SMEs was the Critical Decision Method. We feel that this approach served us quite well, allowing us to capture and organize the relevant details of the SME's incident in enough detail to recreate it in the simulation. This technique is typically employed in an effort to study naturalistic decision making, and thus, is well-suited to the up-front analytical activities required when building a system used, in part, to train decision making using simulation of a real world incident. As with any technique, it requires a bit of practice, and we feel our interviewing skills increased with each successive SME interview. We would recommend this technique to anyone who is working with SMEs to understand and recreate a real world event.

Human Performance Model Development. Another success involved the design of the human performance model that drove the actions of the synthetic players in the scenario. Members of the team have a fairly long history of developing IMPRINT-based human performance models and integrating them with other simulation systems. A valuable lesson, which gets reinforced from project to project, is to create models that are no more complex that necessary to serve their purpose. In this case, our purpose was not to create detailed cognitive and perceptual models of the synthetic WDs, which would ultimately result in somewhat variable behavior from trial to trial. Not only would such detailed models be time-consuming and costly to create and maintain, but they would actually be *counter-productive* to our purposes of supporting the training environment. In order to consistently meet the training objectives, we wanted the behavior of our synthetic players to be predictable, both in the types of actions they take (including planned errors) and the timing at which they occur. Thus, we intentionally built relatively simple task network models to ensure that WD actions (which served as the "stimuli" in our training scenario) occurred as planned to support the training objectives. This eliminated the need for complex task networks, release conditions, and tactical decision logic in the model and the associated expense of analyzing WD tasks in enough detail to represent these complex decisions and actions in the model.

Although the version of IMPRINT used in this effort (*CART 1.20a* --a custom version of IMPRINT developed for the CART program and corresponding roughly to IMPRINT Version 6) supported the modeling needs of this effort quite well, future instantiations would benefit from migrating to the IMPRINT Pro version. During the course of this development effort the Army Research Laboratory released a beta version of the next generation of IMPRINT, IMPRINT Pro. In IMPRINT Pro, the database and coding environment was moved from a 16-bit Borland C environment to a 32-bit Microsoft C Sharp environment. This upgrade has increased the stability and supportability of the tool, increased the power of the programming language, and improved run times. It also provides a means of running in real time that does not rely on HLA time management, thereby increasing the speed and efficiency model runs in an integrated simulation environment.

Speech Recognition and Synthesis. The speech recognition/synthesis capability posed, by far, the most significant technical challenge to the project, and though satisfied with the end result, we acknowledge that the system falls well short of supporting the range of speech interactions that live operators engage in routinely. The notion of developing a voice interaction system that could maintain situation awareness, understand and correctly interpret anything that a live SD might say, and respond with a context-appropriate synthetic verbal response was well-beyond the scope of this effort; and perhaps beyond the ability of today's technology. Rather, we had to adopt an approach commonly used in design of today's voice interaction systems (e.g., airline reservation systems, automated customer services for banking), in which developers create a script involving the most likely verbal interactions that must be supported for a given application. As one voice interaction researcher/author describes it "voice interaction design is, in very large part, the provision and enablement of scripts in the classic artificial-intelligence sense" (Harris, 2005, p. 427), where the system assembles pieces of a pre-defined script in such a way as to support the user's goals. In our view, the reliance upon scripts will never allow full conversational speech between live and synthetic players; however, scripted speech is what today's speech system technology supports.

The size of the script is driven by a number of factors, starting with the desired functionality requirements of the system that the voice interface supports. The number and nature of these requirements will impact the scope of the voice interaction system, both in terms of the conversational breadth (number of topics it must support) and depth (ability to maintain context and provide significant content in conversations on a topic) it must support; and as the scope increases, so do the effort and costs associated with developing the system. For our application and scenario, we were fortunate that the breadth and depth of the script could be kept small enough to be accomplished within the project budget. More than one AWACS SME told us that, if the WDs are performing their duties correctly, the SD will have little, if anything, to say. We could therefore focus our speech interaction development around scenario errors that our synthetic WDs were programmed to commit. Although we needed to create a large number of speech phrases (radio and intercom calls) that the live SD would normally hear to maintain SA during the scenario, we could limit the number SD-initiated voice interactions the system would need to support to those related to "corrective action" regarding a WD error.

Our advice to developers attempting to integrate voice recognition and synthesis into a larger scenario simulation is to set realistic requirements concerning the breadth and depth of voice interactions the system is to support. It is important to understand that supporting even a minimal level of conversation in which context is to be maintained can be quite challenging. Further, in such a system, complexity can grow exponentially with each potential speech component (content and/or phrasing) expected to be understood and responded to by the speech recognition/synthesis system. It is not an undertaking that should be entered into lightly.

Final Thoughts

The CAS prototype demonstration described here represents only one of a myriad of potential applications of human behavior representations across a host of domains. As our modeling, speech, and visualization technologies continue to advance, the quality (realism) of these human behavior representations will continue to progress, and their potential applications will continue to expand. As with advancing technologies in any domain, there will always be differences between what is possible and what is feasible and affordable. Pursuing both fronts (i.e., exploring and discovering new HBR tools and technologies and finding creative, economical ways to apply existing HBR technologies to immediate applications) is essential to advancing the state of the art. Through projects like the CAS prototype demonstration, AFRL seeks to continue advancing the science of human behavior representation and its application to critical causes including systems acquisition, human performance research, and training.

REFERENCES

Brett, B. E., Doyal, J. A., Malek, D. M., Martin, E. A., Hoagland, D. G., & Anesgart, M. N. (2002). *The Combat Automation Requirements Testbed (CART) Task 5 Interim Report: Modeling a Strike Fighter Pilot Conducting a Time Critical Target Mission*. U.S. Air Force Research Laboratory Technical Report AFRL-HE-WP-TR-2002-0018. Wright-Patterson Air Force Base, OH.

Cannon-Bowers, J. A., & Bell, H. H. (1997). Training Decision Makers for Complex Environments: Implication of the Naturalistic Decision Making Perspective. In C. E. Zsambok & G. Klein (Eds.), *Naturalistic Decision Making* (pp 99-110). Mahwah New Jersey: Lawrence Erlbaum Associates, Inc.

DOD 5000.59-P (1995). *Modeling and Simulation (M&S) Master Plan*, Defense Modeling and Simulation Office (DMSO), Alexandria, VA..

Doyal, J. A., Brett, B. E., Martin, E. A., & Barbato, G. J. (2007). *The Combat Automation Requirements Testbed (CART) Program.* Final report: Sept 1998 – Dec 2006. U.S. Air Force Research Laboratory Technical Report. AFRL-HE-WP-TR-2007-0007. Wright-Patterson Air Force Base, OH.

Gluck, K. A., Pew, R. W., & Young, M. J. (2005). Background, Structure, and Preview of the Model Comparison. In K. Gluck & R. Pew (Eds.), *Modeling Human Behavior with Integrated Cognitive Architectures* (pp3-12). Mahwah New Jersey: Lawrence Erlbaum Associates, Inc.

Harris, R. A., (2005). Voice *Interaction Design: Crafting the New Conversational Speech Systems*. San Francisco, CA. Morgan Kaufmann Publishers –an Imprint of Elsevier.

Klein, G. A., Calderwood, R., & MacGregor, D. (1989). Critical decision method for eliciting knowledge. *IEEE Transactions on Systems, Man and Cybernetics*, 19(3), (pp 462-472).

Zsambok, C. E., (1997). Naturalistic decision making: Where are we now? In C. E. Zsambok & G. Klein (Eds.), *Naturalistic Decision Making* (pp 3-16). Mahwah New Jersey: Lawrence Erlbaum Associates, Inc.

Appendix 1: System Requirements for AWACS CAS Prototype Demonstration

Appendix 1 – Systems Requirements for AWACS CAS

Component	**Requirement Number**	**Requirement Description**
Weapons Director Display Content Processor (WDDCP)	1	The ACAS demonstration system shall consist of a Weapons Director Display Content Processor (WDDCP).
	1.1	The WDDCP shall be built using the OpenMap package / environment.
	1.2	The WDDCP shall be capable of supporting two independent Weapons Director Displays.
	1.2.1	Each WD Display shall have a button area.
	1.2.1.1	The button area of the WD Display shall be capable of displaying the switch state of select buttons.
	1.2.2	Each WD Display shall have a map / track area.
	1.2.2.1	The map area of the WD Display shall be capable of displaying an irregular shaped Area of Responsibility (AOR).
	1.2.2.1.1	The WD Display AOR shall include multiple irregular shaped polygons and appropriate text for the Nellis training range.
	1.2.2.2	The map area of the WD Display shall be capable of a map zoom.
	1.2.2.3	The map area of the WD Display shall be capable of a map pan.
	1.2.2.4	The map area of the WD Display shall be capable of changing to a commanded center latitude / longitude.
	1.2.2.5	The map area of the WD Display shall be capable of drawing tracks.
	1.2.2.5.1	The track data shall be refreshed as updated track data is received.
	1.2.2.5.2	The WD Display shall be capable of displaying the associated symbology for tracks in the map area at their current position.
	1.2.2.5.3	The WD Display shall be capable of displaying the appropriate associated data for a track in the map area. Associated data, at a minimum, will include heading, altitude, and speed.
	1.2.2.5.4	The tracks in the map area of the WD Display shall be capable of displaying Tag information
	1.2.2.6	The map area of the WD Display shall be capable of showing a track history for each track.
	1.2.2.7	The map area of the WD Display shall be capable of drawing multiple sized "bullseye" points (or symbols) and associated text (if appropriate).
	1.2.3	Each WD Display shall have a functional "free text message" area.

Component	**Requirement Number**	**Requirement Description**
	1.2.4	The map area of each WD Display shall be capable of adding and displaying "special markers" as the result of the JTIDS 3.5 messages.
	1.3	The WDDCP shall support/receive Track Data via an interface with the Track Generator software.
Weapons Director Display Content Processor (WDDCP) – Continued -	1.3.1	The WDDCP shall read / store / process the Track Data.
	1.4	The WDDCP shall support/receive Command Data via an interface with the Track Generator software.
	1.4.1	The WDDCP shall read/store/process the Command Data.
	1.5	The WDDCP shall receive Free Text Message area data via an interface with the Track Generator software.
	1.6	The WDDCP shall provide an interface to allow a human (SD) to interact with a Weapon Director display.
	1.7	The WDDCP shall be capable of resetting to the start of the scenario without restarting the software.
Track Generator Middleware (TGM) / ("ACAS Tools")	2	The ACAS demonstration system shall include the Track Generator Middleware (TGM) component.
	2.1	The TGM shall be built using the Virtual Warrior TCT Tools software as the baseline.
	2.2	The TGM shall be capable of generating "tracks" for simulated AWACS radar scopes.
	2.2.1	The TGM shall support at least two independent radar scopes
	2.2.1.1	The TGM radar scope shall handle a minimum of 100 active tracks.
	2.2.3	The TGM shall model the entities required for the demo scenario.
	2.2.3.1	The TGM shall model a two-ship of F16CJs ("Bronco-41").
	2.2.3.2	The TGM shall model a four-ship of F15Es ("Claw-21").
	2.2.3.3	The TGM shall model a two-ship of F15Es ("Sword-31").
	2.2.3.4	The TGM shall model two KC135s ("Exxon-23 and Exxon-32").
	2.2.3.4.1	The TGM shall send the location data for the KC135 aircraft to the HPM at approximately 12 second intervals.
	2.2.3.5	The TGM shall model an AWACS ("Mojo") aircraft.
	2.2.3.6	The TGM shall model a JSTARS ("Sabre") aircraft.

Component	Requirement Number	Requirement Description
	2.2.3.7	The TGM software shall model track location, altitude, identifier, (callsign mapping), track type (aircraft type), friend/foe/unknown, airspeed, and heading.
	2.2.4	The TGM software shall interface with the Weapons Directors HPM.
	2.2.4.1	The TGM software shall receive WD Display Command Manipulation Data from the HPM.
	2.2.4.1.1	The TGM software shall interpret and store the HPM command manipulation request to determine the current screen content bounds / settings for each WD position.
	2.2.4.2	The TGM software shall receive Free Text Message area data from the HPM.
	2.2.5	The TGM software shall interface with the Weapons Director Display Content Processor (WDDCP).
Track Generator Middleware (TGM) / TCT Tools -- Continued	2.2.5.1	The TGM shall support/send the WD Display Command Manipulation Data via an interface to WDDCP software.
	2.2.5.1.1	The TGM shall write the Command Data upon receipt of the Command data from the HPM (i.e., relay as received).
	2.2.5.2	The TGM software shall pass Free Text Message area data to the WDDCP when received from the HPM.
	2.2.5.3	The TGM shall support/send the Track Data via an interface to Weapon's Director Display Control Processor software.
	2.2.5.3.1	The TGM shall write the Track Data.
	2.2.6	The TGM shall model ground entities.
	2.2.6.1	The TGM shall be used to generate a column of tanks and their movement, a stationary SOF position ("Tiger-13"), and a stationary SA-8 position.
	2.2.6.2	The TGM shall produce DIS Entity State PDUs for the ground entities.
	2.3	The TGM shall support a Virtual Character Display (VCD).
	2.3.1	[DELETED]
	2.3.2	The TGM VCD shall load a 3-D environment which contains an AWACS-like interior.
	2.3.2.1	The TGM VCD AWACS-like interior shall include at least two Weapons Director (WD) positions/work areas.
	2.3.3	The TGM VCD shall support the WD consoles drawn with textures.
	2.3.3.1	The TGM VCD WD console' textures shall be updated / refreshed approximately every 12 seconds.
	2.3.4	The TGM VCD shall be capable of drawing at least two WD characters.

Component	**Requirement Number**	**Requirement Description**
	2.3.4.1	The WD characters shall have the ability to point toward various areas of the WD console.
	2.3.4.2	The WD characters shall have the ability to turn towards their console, the other WD, and the SD (i.e., towards the live viewer position).
	2.3.4.3	The WD characters shall have the ability to speak.
	2.3.4.4	The WD characters shall support reasonable lip-synced speech (i.e., have expressive faces).
	2.3.4.5	The WD characters shall have flight suits as uniforms with appropriate patches.
	2.3.4.6	The WD characters shall have David Clark headsets with mics.
	2.3.5	The TGM VCD shall support the capability to "play" voice messages from aircraft/tracks (check-in, etc).
	2.4	The TGM software shall include an ACAS aircraft capability which will be an enhanced version of the TCT Tool Attack AC capability.
	2.4.1	The TGM ACAS aircraft capability shall model 25 aircraft.
	2.4.2	The TGM ACAS aircraft capability shall provide the ability to specify the aircraft type.
Track Generator Middleware (TGM) / TCT Tools -- Continued	2.4.3	The TGM ACAS aircraft shall provide initialization/support of at least 3 waypoints for each aircraft.
	2.4.4	The TGM ACAS aircraft shall begin a CAP orbit upon reaching the last waypoint.
	2.4.5	The TGM ACAS aircraft shall provide a capability to retask an aircraft based on a request from the WDs.
	2.4.6	The TGM ACAS aircraft shall begin a CAP orbit upon arrival at retask point
	2.4.7	The TGM ACAS aircraft shall provide the ability to "say" and "respond" to key events along the route
	2.4.7.1	The TGM ACAS aircraft shall be capable of initiating a "check-in" voice interaction
	2.4.7.2	The TGM ACAS aircraft shall be capable of responding to a post "check-in" handoff voice interaction
	2.4.8	The TGM ACAS aircraft shall provide the ability to "contact" the JTAC when tasked to a target.
	2.4.8.1	The ACAS aircraft shall simulate sending the contact message to the JTAC when it reaches the destination point.
	2.4.8.2	The ACAS aircraft shall process a simulated message from the JTAC indicating whether the aircraft's weapon load is sufficient for the target.

Component	**Requirement Number**	**Requirement Description**
	2.4.8.2.1	The ACAS aircraft shall contact WD3 with a "bad load" message if it receives an insufficient load response from the JTAC.
	2.4.8.3	The ACAS aircraft shall accept and act on a retask message from the JTAC.
	2.4.9	The TGM shall send the location data for all ACAS aircraft to the HPM at approximately 12 second intervals.
	2.4.10	The TGM shall produce DIS Entity State PDUs for all aircraft entities.
	2.5	The TGM shall support a Data Collection capability
	2.5.1	The TGM Data Collection data shall record all received key events include the timestamp, key event identifier, and any relevant meta data.
	2.5.2	The TGM Data Collection data shall record all voice message traffic.
	2.6	The TGM shall support simulated JTIDS 3.5 messages.
	2.7	The TGM shall support multiple communication channels.
	2.7.1	The communication channels shall include AWACS internal Intercom, SATCOM, and Radio (upto 3 freq). AWACS Intercom includes Net 1 (Flight Deck/MCC, MX coordination), Net 2 (Weapons), Net 3 (Surveillance/ECO), and "Net 4" (internal voice),
	2.8	The TGM shall create a GUI window at startup.
	2.8.1	The TGM GUI shall have buttons for simulation start/re-start, pause/resume, and shutdown.
	2.8.2	The TGM GUI shall have a button to spawn/play a PowerPoint presentation for a pre-brief capability.
Track Generator Middleware (TGM) / TCT Tools -- Continued	2.8.2.1	A pre-brief presentation shall be developed for inclusion at the start of the demo.
	2.8.2.2	The PowerPoint briefing shall be self paced.
	2.8.2.3	The content of the PowerPoint briefing shall include context for the mission, roles of the individual characters (including the SD), intended purpose of the simulation demonstrator, and the general operation.
	2.8.2.4	The content in the PowerPoint briefing shall be replaceable by a user without impacting other components of the demo system.
	2.8.3	The TGM GUI shall provide a capability to set a CBT mode on or off.
	2.8.3.1	The TGM shall support CBT-type "interruptions" when CBT mode is ON.
	2.8.4	The TGM GUI shall display simulation status, simulation time, and elapsed simulation time.

Component	**Requirement Number**	**Requirement Description**
	2.9	The TGM shall provide an event based capability to generate scheduled messages / external source message stimuli.
	2.9.1	The TGM event queue will include time of event, id of originator, and destination information.
	2.10	The TGM shall be capable of resetting to the start of the scenario without restarting the software.
Speech Processing	3	The ACAS demonstration system shall consist of a Speech Processing (SP) component.
	3.1	[DELETED]
	3.2	The Speech Processing component shall provide speech synthesis and speech recognition capabilities.
	3.3	The speech synthesis shall provide a capability to generate audio files for phrases spoken by the synthetic players in the system.
	3.4	The speech synthesis shall provide a capability to generate at least 12 unique voices.
	3.4a	The speech recognition shall provide a capability to recognize the live Senior Director (SD).
	3.5	The speech recognition shall provide a capability to classify a message from the live SD as being directed to a particular WD (WD2 / WD3).
	3.6	The speech recognition capability shall recognize multiple distinct speech phrases (i.e., grammars mapping to enumerations) from the SD as defined in the ICD.
	3.6a	The speech recognition capability shall support partially recognized phrases and report them to the HPM as partial inputs.
	3.7	The speech synthesis capability shall support multiple communication channels.
	3.7.1	The speech synthesis capability shall degrade voice communications which occur on the SATCOM and Radio frequencies.
	3.8	The Speech Processing component shall be capable of resetting to the start of the scenario without restarting the software.
Weapons Director (WD) Human Performance Model (HPM)	4	The ACAS demonstration system shall consist of a Weapons Director Human Performance Model (WDHPM).
	4.1	The WDHPM shall be built using the IMPRINT environment.
	4.2	The WDHPM shall model the appropriate AWACS internal personnel / positions.
	4.2.1	The WDHPM shall model, at a minimum, two Weapon's Directors positions.

Component	Requirement Number	Requirement Description
	4.2.1.1	The WD2 position shall be modeled as a Check-In / Tanker / High Value Airborne Asset (HVAA) Controller.
	4.2.1.2	The WD3 position shall be modeled as a Dynamic Target (DT) Controller.
	4.2.1.3	The Weapon's Directors positions shall support a limited level of interactions between WDs as documented in the ICDs.
	4.2.2	The WDHPM shall model the Electronic Combat Officer (ECO), Mission Crew Commander (MCC), and the Flight Engineer (FE).
	4.2.2.1	The WDHPM ECO shall simulate receipt of a JTIDS 3.5 message and push the appropriate data (location, symbology, etc) associated with a track (representing the column of tanks).
	4.2.2.2	The WDHPM MCC shall receive and acknowledge a message about fuel state from the FE.
	4.2.2.3	The WDHPM FE shall send a voice message about fuel state to the MCC.
	4.3	The WDHPM shall interface with a human Senior Director (SD) through the speech processing system (speech recognition/synthesis interactions)
	4.4	The WDHPM shall model the appropriate external personnel/systems that interact with the AWACS.
	4.4.1	The WDHPM shall model a JTAC position (external to the AWACS).
	4.4.1.2	The JTAC position shall be capable of "initiating" a request for support to the ECO.
	4.4.1.3	The JTAC position shall be capable of receiving a 'contact' message from ACAS aircraft.
	4.4.1.3.1	The JTAC position shall provide the capability to determine whether the aircraft that has contacted it contains the proper weapon load.
	4.4.1.3.2	The JTAC position shall support an interaction with an ACAS aircraft to simulate response to the 'contact' event.
	4.4.1.3.2.1	The JTAC 'contact' response shall include an indication of satisfactory or unsatisfactory weapon load.
	4.4.1.4	The JTAC position shall be capable of retasking the ACAS aircraft.
	4.5	The WDHPM shall interact with ACAS aircraft (i.e., tracks) in the environment.
	4.5.1	The WDHPM shall provide the capability to acknowledge an ACAS aircraft "check-in".
	4.5.2	The WDHPM shall provide the capability to handoff aircraft (to the other WD or another external entity).
	4.5.3	The WDHPM shall provide the capability to perform voice queries (as defined in the ICD).
	4.5.4	The WDHPM shall provide the capability to perform voice responses (as defined in the ICD).

Component	**Requirement Number**	**Requirement Description**
Weapons Director (WD) Human Performance Model (HPM) – Continued --	4.5.5	The WDHPM shall be capable of retasking the ACAS aircraft (non-voice).
	4.5.6	The WDHPM shall provide the capability to "tag" a track.
	4.5.7	The WDHPM shall be capable of computing Bearing, Range, Altitude (BRA) from an ACAS aircraft to a KC135.
	4.5.8	The WDHPM shall be capable of computing a "contact" location between an ACAS aircraft and a target location.
	4.5.9	The WDHPM shall model/store the scenario Bullseye, checkin locations, and other key locations / borders for performing calculations related to key events.
	4.6	The WDHPM shall manage the content of each of the WD Displays.
	4.6.1	The WDHPM shall set the initial center location and map scale on the WD Display.
	4.6.2	The WDHPM shall add any special symbols to the WD Display.
	4.6.3	The WDHMP shall manage the "tagged" aircraft information that appears on the WD Display.
	4.6.4	The WDHPM shall manage the Free Text Message area that appears on the WD Display.
	4.7	The WDHPM shall support multiple communication channels.
	4.7.1	The communication channels shall include AWACS internal Intercom, SATCOM, and Radio (up to 3 freq). AWACS Intercom includes Net 1 (Flight Deck/MCC, MX coordination), Net 2 (Weapons), Net 3 (Surveillance/ECO), and "Net 4" (internal voice),
	4.8	The WDHPM shall monitor the SD voice inputs for corrective actions or queries for information.
	4.8.1	The WDHPM shall be capable of handling partially recognized phrases from the SD.
	4.8.2	The WDPM shall upon receipt of partially recognized phrases query the SD for the missing information.
	4.8.3	The WDHPM shall be capable of a "don't understand" response when asked to do things not required, implemented, or understood.
	4.9	The WDHPM shall perform corrective actions when directed by the SD
	4.9.1	The WDHPM corrective actions shall include interrogation of an ACAS aircraft upon command, redirection (retask of aircraft), and handoff.
	4.10	The WDHPM shall send key events to data collection capability (based on the ACAS ICD).

Component	**Requirement Number**	**Requirement Description**
	4.11	The WDHPM shall manage the creation, movement, and speech of the WD2 and WD3 virtual characters.
	4.12	The WDHPM shall be capable of resetting to the start of the scenario without restarting the software.

Appendix 2. Pre-Mission Briefing, Pop Up Hints, and Post-Mission Briefing Slide

Pre-Mission Briefing Slides

Corrective Action Simulator

-- A Conceptual Demonstration--

--Press "Page Down" on keyboard to start pre-mission briefing

or

--Press "Start / Restart Simulation" button on the Simulation Control Panel to bypass the briefing and start the scenario--

Demonstration Overview

- This demonstration is intended to convey the concept of a "Corrective Action Simulator" (CAS)
- The CAS:
  - Immerses a live operator in a synthetic real-world environment
  - Allows the live operator to observe a team of subordinates performing a real-world task
  - Allows the live operator to intervene in the scenario, initiating corrective action when a subordinate is observed to make an error (of omission or commission)

Demonstration Overview (continued)

- Benefits of a Corrective Action Simulator
  - Provides immersive setting for training
  - Can be used to convey actual real-world events or hypothetical scenarios
  - Provides an environment that allows the study and/or training of naturalistic decision making (i.e., how people make decisions in complex field settings)

Setting

- This demonstration depicts an AWACS environment in which a live Senior Director (SD) observes two Weapons Directors (WDs) performing a mission in support of dynamic targeting.
- "WD 2" is charged with checking aircraft in as they enter the operating area. "WD 3"s role is to assign strike aircraft to dynamic targets as they appear.
- Assuming the role of Senior Director, the live player (you) should observe these individuals, paying particular attention to their verbal interactions with each other and with the aircraft they are controlling.

Instructions

- Review the following Mission Overview slides to become acquainted with the players/roles in the scenario
- When ready to begin the scenario, select "Start / Restart Simulation" on the Simulation Control Panel
- Observe synthetic operators performing in the mission environment

Instructions (continued)

- If you observe one of the WDs perform an incorrect action or fail to perform an action when one is appropriate, verbally suggest a corrective action to the given WD
- To speak with a WD:
  - press and hold the microphone footswitch
  - address the given WD and immediately state the desired action*
  - release the footswitch when you have completed the verbal statement

e.g.,

[press footswitch] Speak, "WD 2, SD. Reset Bronco." [release footswitch]

- Follow any on-screen "hints" that may appear during the scenario

* If necessary, refer to handout showing supported speech interactions

Mission Overview

- Medium-sized force training in the Nellis ranges (ranges modified for demo)
- No expected air-to-air threat
- Mission: On-call CAS and TST
- Beginning mission time: 0850Z

Mission Overview (continued)

- Assets
  - Strike
    - 6xF-15E
    - 2xF-16CJ
  - DCA
    - 4xF-15C
  - C2
    - E-3A
  - ISR
    - Joint STARS
  - A/R
    - 2xKC-135R
  - Ground
    - 1xJTAC

ATO

```
TASKUNIT/552ACS/ICAO:KTIK//
AMSNDAT/2001/-/-/-/AEW/-/-/DEPLOC:KLSV/ARRLOC:KLSV//
MSNACFT/1/ACTYP:E3B/MAGIC35/-/-/101/22001/32001//
AMSNLOC/010200ZNOV/011100ZNOV/AWACS/280//
CONTROLA/CRC/NELLIS/PFREQ:122.0/SFREQ:331.0/NAME:CHECKIN//
AMPN/CONTROL TASKUNIT: 728 ACS//
ASACSDAT/AWAC/MOJO/E3B/AEW/-/-/PFREQ:BEIGE5/SFREQ:WHITE5//
7CONTROL
/MSNNO /ACSIGN      /NO /ACTYPE  /MSNTY  /TOSTA  /RIP
/2002    /CYLON11    /   4  /AC:F15C  /DCA      /061000Z/3717N11455W
/2003    /BRONCO21/   2  /AC:F16C   /AI         /061000Z/3717N11455W
/2004    /CLAW21     /   4  /AC:F15E   /AI         /061000Z/3717N11455W
/2005    /SWORD31  /   2  /AC:F15E   /AI         /061000Z/3717N11455W
/2006    /SABRE11   /   1  /AC:E8C     /C2         /061000Z/3717N11455W
/2007    /EXXON32   /   1  /AC:KC135 /AR        /061000Z/3717N11455W
/2008    /EXXON33   /   1  /AC:KC135 /AR        /061000Z/3717N11455W

TASKUNIT/58 FS/ICAO:KVPS//
AMSNDAT/2002/-/-/-/DCA/-/-/DEPLOC:KLSV/ARRLOC:KLSV//
MSNACFT/4/ACTYP:F15C/CYLON11/ A4A2S2W1/-/101/22002/32002//
AMSNLOC/010830ZNOV/011100ZNOV/CAP/250//
CONTROLA/AWAC/MOJO/PFREQ:BEIGE5/SFREQ:WHITE5/-//
AAR/EXXON32/2007/210/0945/50K/GREEN3/BLUE3/-

TASKUNIT/4 FS/ICAO:KHIF//
AMSNDAT/2003/-/-/-/AI/-/-/DEPLOC:KLSV/ARRLOC:KLSV//
MSNACFT/2/ACTYP:F16C/BRONCO21/ 2A88X2G31 /-/101/22003/32003//
AMSNLOC/010900ZNOV/011030ZNOV/81J/200//
CONTROLA/AWAC/MOJO/PFREQ:BEIGE5/SFREQ:WHITE5/-//
AAR/EXXON32/2007/210/0855/10K/GREEN3/BLUE3/-
```

ATO - continued

```
TASKUNIT/334 FS/ICAO:KGSB//
AMSNDAT/2004/-/-/-/AI/-/-/DEPLOC:KLSV/ARRLOC:KLSV//
MSNACFT/4/ACTYP:F15E/CLAW21/ 6G31X6C103/-/101/22004/32004//
AMSNLOC/010910ZNOV/011045ZNOV/81F/210//
CONTROLA/AWAC/MOJO/PFREQ:BEIGE5/SFREQ:WHITE5/-//
AAR/EXXON33/2007/200/1000/50K/GREEN3/BLUE3/-

TASKUNIT/335 FS/ICAO:KGSB//
AMSNDAT/2005/-/-/-/AI/-/-/DEPLOC:KLSV/ARRLOC:KLSV//
MSNACFT/2/ACTYP:F15E/SWORD31/ 4G12X3C87X2AW2 /-
/101/22005/32005//
AMSNLOC/010910ZNOV/011045ZNOV/81F/210//
CONTROLA/AWAC/MOJO/PFREQ:BEIGE5/SFREQ:WHITE5/-//
AAR/EXXON33/2007/200/1020/50K/GREEN3/BLUE3/-

TASKUNIT/116 ACW/ICAO:KWRB//
AMSNDAT/2006/-/-/-/C2/-/-/DEPLOC:KLSV/ARRLOC:KLSV//
MSNACFT/1/ACTYP:E8C/SABRE11/-/-/101/22006/32006//
AMSNLOC/010800ZNOV/011100ZNOV/AOR/200//
CONTROLA/AWAC/MOJO/PFREQ:BEIGE5/SFREQ:WHITE5/-//

TASKUNIT/145 ARS/ICAO:KLCK//
AMSNDAT/2007/-/-/-/AR/-/-/DEPLOC:KLSV/ARRLOC:KLSV//
MSNACFT/1/ACTYP:KC135/EXXON33/BEST/-/101/22007/32007//
AMSNLOC/010800ZNOV/011100ZNOV/TANKER1/210//
REFTSK/BOM/KLBS:060/-/-//
5REFUEL
/MSNNO /RECCS /NO/ACTYPE /OFLD /ARCT /SEQ /TYP /ARS
/2003 /BRONCO21 / 2/AC:F16C /KLB:010/060855Z/ -/A:JP8/BOM//
/2002 /CYLON11     / 4/AC:F15C /KLB:050/06945Z/ -/A:JP8/BOM//
CONTROLA/AWAC/MOJO/PFREQ:BEIGE5/SFREQ:WHITE5/NAME:CHECKIN//

TASKUNIT/166 ARS/ICAO:KLCK//
AMSNDAT/2008/-/-/-/AR/-/-/DEPLOC:KLSV/ARRLOC:KLSV//
MSNACFT/1/ACTYP:KC135/EXXON33/BEST/-/101/22008/32008//
AMSNLOC/010800ZNOV/011100ZNOV/TANKER1/200//
REFTSK/BOM/KLBS:100/-/-//
5REFUEL
/MSNNO /RECCS /NO/ACTYPE /OFLD /ARCT /SEQ /TYP /ARS
/2004 /CLAW21    / 4/AC:F15E /KLB:050/061000Z/ -/A:JP8/BOM//
/2005 /SWORD31 / 2/AC:F15E /KLB:050/061020Z/ -/A:JP8/BOM//
CONTROLA/AWAC/MOJO/PFREQ:BEIGE5/SFREQ:WHITE5/NAME:CHECKIN//
```

SPINS/ACO

STANDARD CONVENTIONAL LOADS (SCL)

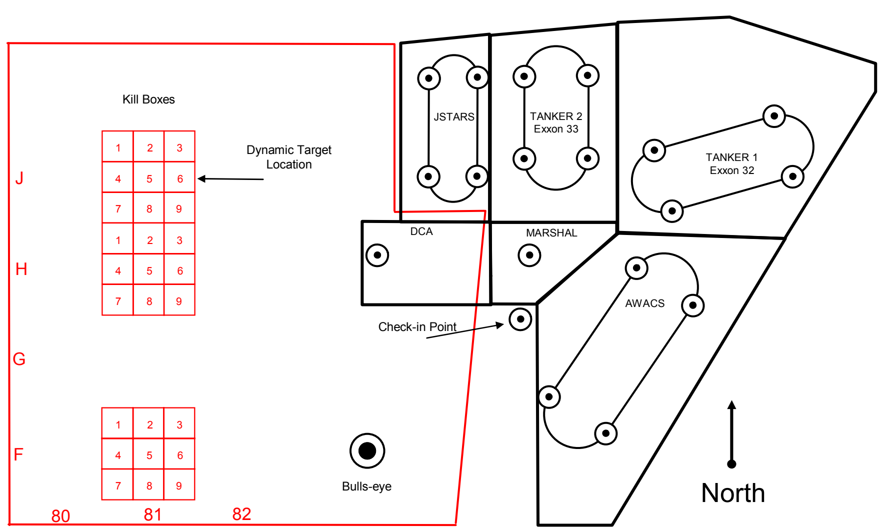
4A2S2W1	4xAIM120	2xAIM7	2xAIM9		
6G31X6C103	6xGBU31A	6xCBU103			
2A88X2G31	2xAGM88	2xGBU31A			
2M82HX3A652W	2xMK82	3xAGM65	4xAIM9		
4G12X3C87X2A2W2	4xGBU12	3xCBU87	2xAIM120	2xAIM9	

BULLSEYES

POSITION	NAME
N 36 56X W 115 27	BULLSEYE

MARSHALL PLAN
INGRESS/EGRESS ALTITUDES ARE PREASSIGNED AS FOLLOWS:
BRONCO: 15000
CLAW: 16000
SWORD: 17000
CYLON: FL200B260

Mission Area



Corrective Action Simulator

-- A Conceptual Demonstration--

--Press the “Start / Restart Simulation” button when ready to start the scenario--

Pop-Up Hints
(shown as necessary during runtime)

Corrective Action Simulator Demo

Corrective Action Required

Sword checked in but failed to report status. In addition, WD2 failed to request Sword's status at check in. This will contribute to a lack of situation awareness.

Corrective Action: Instruct WD2 to Check State on Sword

--Press the "Pause / Resume Simulation" button to continue--

File: ACASError1Brief.ppt

Corrective Action Simulator Demo

Corrective Action Required

WD3 just tasked Sword to contact the JTAC to attack the line of tanks. Sword's actual weapon load is primarily for air-to-air, and thus, Sword is poor choice for tasking against the armored column. Resetting Sword and then retasking Claw to the tanks would be the best option.

Corrective Action: 1) Instruct WD3 to Reset Sword
2) Instruct WD3 to Retask Claw to tanks

--Press the "Pause / Resume Simulation" button to continue--

File: ACASError2Brief.ppt

Post-Mission Results Briefing
(Example)

Corrective Action Simulator Demo

Demo Scenario Complete

Results:

Objective 1 - Maintaining SA:	Challenge: Detecting the absence of Sword state data	
Result 1:	Absence of data detected and corrected in a timely manner	√
Objective 2 – Knowledge Application:	Challenge: Achieving an appropriate weapon/target pairing	
Result 2:	Sword inappropriately tasked against tank column	X

File: ACASScenarioEndBriefGB.ppt

ACRONYM LIST

AETC	Air Education and Training Command
AFRL/RH	Air Force Research Laboratory, Human Effectiveness Directorate
AFRL/RHCP	AFRL/RH's Warfighter Interface Division, Collaborative Interfaces Branch
AMBR	Agent-based Modeling and Behavior Representation
AOR	Area of Responsibility
APC	Armored Personnel Carrier
ATO	Air Tasking Order
AWACS	Airborne Warning And Control System
CART	Combat Automation Requirements Testbed
CAS	Corrective Action Simulator
CBT	Computer Based Training
CDM	Critical Decision Method
COM	Component Object Model
CRC	Control and Reporting Center
DCA	Defensive Counter Air
DIS	Distributed Interactive Simulation
DMSO	Defense Modeling and Simulation Office
DOD	Department of Defense
GUI	Graphical User Interface
IFF	Identification Friend or Foe
IMPRINT	Improved Performance Research Integration Tool
JTAC	Joint Terminal Attack Controller
HBR	Human Behavior Representation
HLA	High Level Architecture
HPM	Human Performance Model
JSTARS	Joint Surveillance Target Attack Radar System
MCC	Mission Crew Commander
NDM	Naturalistic Decision Making
PC	Personal Computer
SAIC	Science Applications International Corporation
SAM	Surface to Air Missile
SD	Senior Director
SDK	Software Development Kit
SA	Situation Awareness
SME	Subject Matter Expert
SPINS	Special Instructions
USAF	United States Air Force
WD	Weapons Director
WDDCP	Weapons Director Display Content Processor