REPORT DOCUMENTATION PAGE

AFRL-SR-BL-TR-00-

0604

1. AGENCY USE ONLY *(Leave blank)*

2. REPORT DATE

3. RE... ...D DATES COVERED
Final - 1 September 1998 - 1 June 1999

4. TITLE AND SUBTITLE
~~STTR-I Real Time Intelligent Coaching for Command and Control~~

5. FUNDING NUMBERS
F49620-98-C-0055

6. AUTHOR(S)
Dr. Daniel Serfaty

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Aptima Inc
600 West Cummings Park, Suite 3050
Woburn, MA 01801

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NL
801 North Randolph Street, Room 732
Arlington, VA 22203-1977

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*
We produced a command and control coaching model.and wrote a mission needs statement for AWACS. We also created a storyboard description of INTACT. We sucessfully demonstrated a prototype INTACT supporting real time operator performance during a DDD simulation.

14. SUBJECT TERMS
Command and Control,

15. NUMBER OF PAGES
99

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclas	Unclas	Unclas	

Topic STTR AF098-008 Phase I Final Report

INTACT: Intelligent Tactical Coaching Tool

Submitted By:

Aptima, Inc.
600 West Cummings Park
Suite 3050
Woburn, MA 01801

Advanced Human Resource Project (AHRP) Laboratory
University of Georgia
Athens, GA 30602

Table of Contents

List of Figures

List of Tables

1. INTRODUCTION

Today's military command and control missions are information rich, complex processes executed by networked individuals, supported by highly sophisticated hardware, all functioning in dynamic and uncertain environments. They require extensive communications, coordination, synchronization, and information management with the human decision-maker in a central role within these organizations. These complex multi-task work environments are the expected standard for command and control in the coming years. Human factors specialists, however, have long been concerned about the operators' ability to reliably handle several tasks simultaneously.

As the military's command and control systems are upgraded, advanced technical systems intended to improve overall performance will be added on top of today's already complex systems, further requiring the human operators to "juggle" multiple tasks, switching rapidly from one task to another in order to respond to changing situations without losing track of overall priorities. The ultimate implications of these rapid advances in information processing and network technology for the future of warfare are far from obvious. What is clear, however, is that there is a need to provide intelligent tools to support the operator in these complex, multi-task environments in order to maintain acceptable performance levels.

Current trends in the U.S. Armed Forces may be **increasing** the operational challenges faced by the warfighter. Increased turnover within the military requires that more people be trained to perform these operational missions. At the same time, however, budget cuts are forcing a reduction in the time and money available for training. As a result, the number of qualified and available warfighters is decreasing, resulting in more remote duty responsibilities for the remaining warfighters. Air Force Airborne Warning and Control System (AWACS) crews provides an excellent example of this problem. Currently there are so few qualified mission crews that operators can expect to be away from home up to 200 days per year.

Our Phase I effort was specifically intended to simultaneously address the combined problems of an increase in the complexity of command and control tasks and a decrease in the resources available for training operators to perform these tasks. Our ultimate goal is to develop a tool that will support the warfighter functioning in a multi-task command and control work environment as a means to enhance operational performance. In addition, we intend to provide innovative methods to improve the training process, decreasing the time necessary to turn students into qualified operators in these complex environments. In Phase I, we developed a theory-based model of coaching strategies to support command and control decision making to address specified operational needs. Our proposed coaching concepts were demonstrated in the form of a real-time intelligent coach—Intelligent Tactical Coaching Tool or INTACT. This coach was demonstrated in Phase I using a command and control mission simulation as a testbed—the Distributed Dynamic Decisionmaking (DDD) team-in-the-loop simulation that simulates the tasks of an AWACS weapons control team.

INTACT is envisioned as an **embedded tool intended to reduce the workload burden associated with complex, information-rich, multi-task operational environments**. INTACT is also viewed as a tool capable of being used to **aid in the training of new warfighters by providing an embedded coach for simulation based training**. The coach is intended to ensure that a student is able to complete simulated mission tasks correctly, even when unsupervised during extracurricular practice sessions. INTACT is viewed and designed as both an operational and training aid, **supporting the goal of "fight like we train"** by providing a tool that can be used in both settings.

Our multidisciplinary INTACT development team blends specialized expertise in command and control decision analysis, decision modeling, and model-based training from a small business (Aptima, Inc.) with innovative work in intelligent coaching strategies from a university research team (University of Georgia). The team's strengths include an extensive background in command and control modeling, the development of intelligent agents for command and control decisionmaking, innovative ideas for communicating the results of decision models to human decision makers, creation of training programs, and a proven team-in-the-loop simulation test environment with associated performance measures (the DDD) for evaluating intelligent coaching ideas. We are especially well qualified to develop and test ideas for supporting and training individual and team decision making in command and control *teams,* based on team models and using the DDD as a testbed to measure performance.

1.1. Overview of INTACT

Our Phase I goal was to create a theory-based prototype of an intelligent coach for command and control. To achieve this goal, we established and followed a systematic development process, comprised of five objectives, described in detail below. We began this process by developing a model of command and control coaching and then selected a suitable platform to apply this model, in the form of a real-time coaching tool. For Phase I, we selected the AWACS E-3 aircraft as our development platform because of it rich command and control environment and our team's previous experience in this area allowed us to leverage other efforts to meet our aggressive goals in Phase I (see Section 1.2). In Phase I we conducted a mission needs assessment of AWACS to identify the operational areas that should be supported by a coach. This needs assessment was tied to our theory-based model to create design specifications for an intelligent coach. These specifications served as the basis for our INTACT prototype that we demonstrated as a tool to support real-time operator performance on the DDD. Finally, we devised an experimental plan to evaluate the overall utility of INTACT.

INTACT is envisioned as an embedded software tool that supports operational performance and training in tactical decision making environments. Figure 1 presents a graphic representation of the conceptual model that serves as the foundation for INTACT. As can be seen, our theory-based **coaching strategies** will interact directly with the operator or user. The key aspect of INTACT is that it is adaptive to both the **operator's dynamic state** (user's actions) and the **tactical environment** (operational situation) based on the outputs of an **intervention trigger**. Once the intervention trigger decides that a mismatch exists between the operator's actions and the tactical environment (i.e., there is an error or problem), INTACT selects a **coaching strategy** to present information to the user based on models and theories of effective coaching.

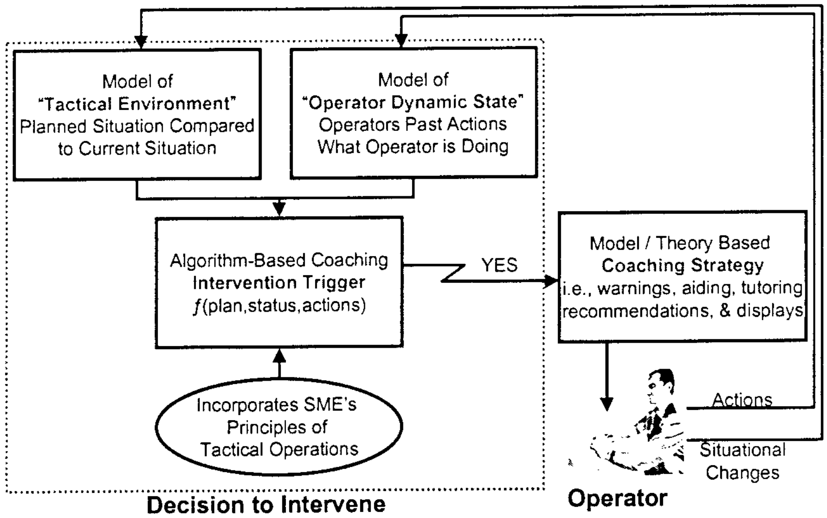


Figure 1. INTACT Application Model

The key components of INTACT are:

Coaching Strategy

Our approach is grounded in a technology that can monitor task and mediation properties in order to help provide levels of operational awareness that match operational demands. Using a framework called the Dynamic Coaching Protocol (DCP) we will develop adaptable software agents that can help guide expertise in military operations. An explicit metatheory will provide coaching principles that can be applied to a variety of operational contexts, whereas the triggering components (below) of the coaching system will be scaled to the operational contexts at hand.

Tactical Environment Model

This model is a representation of how things should be versus how they actually are. For example, expected locations of friendly aircraft, as dictated by battle plans (e.g., air tasking orders), will be compared to current locations. In addition, projected calculations will be made to assess if upcoming actions (e.g., waypoints, targets, tankers) are reachable given current location, aircraft capabilities, and location of action.

Operator Dynamic State Model

This model will assess operator actions in relation to mission objectives. The coach will maintain a set of actions or "checklists" associated with each task of the air battle plan. As the operator performs various mission tasks, the coach will record the operators' actions and relate them to the checklists. This will provide the ability to address errors of omission and reduce the likelihood of issuing a false alarm warning (e.g., if operator already vectored a fighter to a tanker, there is no need to provide a warning that the fighter is behind schedule for refueling).

Intervention Trigger

This is an algorithm that combines the two models described above and triggers the coach to prompt the operator. In addition to the two models, this process will incorporate basic tactical principles to guide the decision to coach. That is, we will use subject matter experts' input to identify reasons for coaching in the real world (i.e., when would they provide assistance if they were looking over another operators shoulder).

1.1.1. *Phase I Objectives and Accomplishments*

In the course of meeting our five Phase I objectives, we developed three major products: a theory-based model of command and control coaching, an operational needs statement for a targeted command and control organization, and a prototype version of INTACT. The Phase I objectives and accomplishments supporting these products are summarized below.

Phase I-Objective 1:	Develop a theory-based model of command and control coaching.
Accomplishment:	Produced a **command and control coaching model**.

In Phase I, we conducted a literature review in the areas of coaching, training, tutoring, and decision aiding for command and control. Based on this review we defined a theoretical approach that bridges instructional technology and decision support approaches to creating intelligent coaching systems. The approach is grounded in a technology that can monitor task and mediation properties in order to help provide levels of operational awareness that match operational demands. The objective was to build a framework, called the Dynamic Coaching Protocol (DCP), that will lead to the development of adaptable software agents that can help guide expertise in military operations. An explicit metatheory provides coaching principles that can be applied to a variety of operational contexts, while the triggering components of the coaching system are scaled to the operational contexts at hand. The ultimate goal is to create a configurable coaching system for specific applications that leverages context-independent theoretical principles.

The conceptual ideas we advanced in Phase I consolidate three areas of research—judgment and decision making, display engineering, and situational awareness—to enhance the operational awareness of military personnel. The coaching system leverages the existing knowledge of military operators by providing graded levels of feedforward and feedback information, allowing the operator to adjust the level and type of information provided by the coach.

Phase I-Objective 2:	Define an operational needs statement for a targeted command and control organization
Accomplishment:	Wrote a **mission needs statement** for AWACS

The mission needs statement provides a mission-based justification for the proposed structure for the intelligent coach. The needs statement presents descriptions of AWACS mission tasks and possible coach functionality to support the operator. To develop the mission needs statement, we conducted interviews with current AWACS tactics instructors from both Tinker AFB and the Fighter Weapons School at Nellis AFB, and worked with a former AWACS Senior Director (SD) who specializes in AWACS training and simulation. In addition, we reviewed training documents such as the Wing Tactics Standards (10-23) and the Weapons Integration and ATO Execution STP Training Guide.

Based on the interviews, there seems to be agreement that the issue of maintaining situation awareness in a high task saturation environment is a key aspect of an AWACS weapons director's (WD) command and control performance. Experts mentioned dealing with the five phases of an intercept, building and maintaining situation awareness, and multi-tasking under conditions of task saturation as mission components that are hard for WDs to learn and perform. Our Phase I effort, therefore, aimed to develop a coach to support these processes.

Phase I-Objective 3:	Convert theory-based model and operational needs statement into requirements for an intelligent coach.
Accomplishment:	Created a **storyboard** description of INTACT.

The two tasks described above, theory-based model development and operational needs definition, were integrated to develop a set of requirements for the intelligent coach. The intent was to document how the theory-based model of C2 training can be applied to the identified operational needs. We identified specific components of the coach and related them to both the model and the operational needs. These requirements were instantiated in annotated diagrams or storyboards. These storyboards present a graphic display of the concepts for the coach and provide written descriptions of the proposed functionality.

Phase I-Objective 4:	Design a conceptual prototype of the intelligent coach.
Accomplishment:	Successfully demonstrated a **prototype INTACT** supporting real time operator performance during a DDD simulation.

In order to demonstrate our intelligent coaching concepts in a command and control mission environment during Phase I, we integrated coaching software with a multi-user simulation package—the Distributed Dynamic Decisionmaking (DDD) command and control team simulation. The DDD is a unique distributed multi-person simulation for understanding command and control issues in a dynamic team environment. Successive DDD generations have demonstrated the simulation's flexibility in reflecting different domains and scenarios to study realistic and complex command and control decision making. Recently, Aptima developed a version of the DDD that simulates the tasks of an AWACS WD team and is resident at Brooks AFB and the Air Force Academy.

For Phase I, we designed a defensive counter air (DCA) scenario for the DDD simulator that required the operator to conduct a defensive counter air engagement. We included specific scenario elements, related to the mission needs statement, that are possible sources for mistakes on AWACS to demonstrate the capabilities of our coaching tool. The scenario was a direct offshoot of our mission needs statement (Objective 2) and is comprised of tasks that an AWACS operator is expected to complete during DCA maneuvers. Associated with these tasks are possible errors that, when they occur during the mission, will serve as trigger events to engage the coaching functionality. Examples of these tasks include ensuring that all friendlies are committed to hostile targets, selection of proper intercept geometry, maintaining matching force numbers, and providing safe passage out of the air battle theater. During the mission, if the operator fails to complete or incorrectly completes a task, a trigger within the DDD alerts the coach software that an intervention is needed. Based on the scenario and type of error, the coach selects an intervention strategy (as defined by the coaching model). The intervention strategy is also determined, in part, by the operational tempo of the mission.

INTACT was developed as an independent software application, designed to be embedded within the DDD simulator. This architecture required that we define a communication protocol between the coach, written in Java, and the DDD, written in C++, using a TCP/IP socket. Of important note, this interchange is based on the XML protocol which allows us to take advantage of emerging web technologies to enhance performance in future iterations. Figure 2 presents a snapshot of the INTACT JavaCoach software imbedded in the DDD interface.

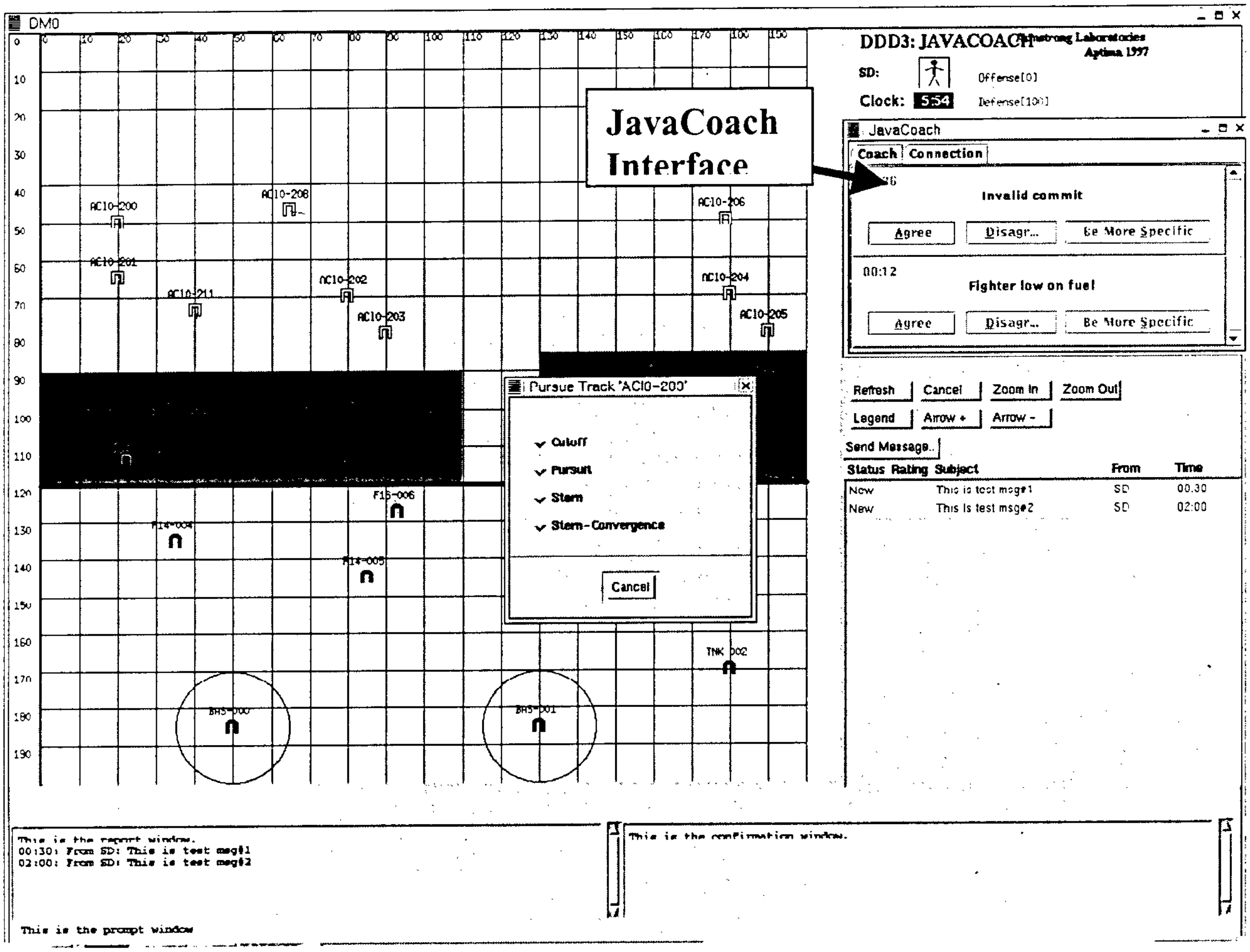


Figure 2. INTACT Prototype: DDD with Embedded JavaCoach

Phase I-Objective 5:	Devise an experimental plan to evaluate various aspects of INTACT and the impact of INTACT on operational performance.
Accomplishment:	Wrote a **Phase II experimental plan** for conducting an evaluation study of INTACT and isolated coaching strategies.

A primary concern when developing any performance support or decision support tool is to assess the practical benefit: "Does it help improve performance?" We have devised a research plan to guide our Phase II experimental evaluation program. The research plan specifies how evaluations will be conducted in Phase II, the hypotheses to be tested, the coaching concepts to be evaluated, the DDD configurations and scenarios to be used in testing those concepts, and the appropriate DDD-based performance measures to be used in the evaluation to test our hypotheses. Beyond the overall performance impact of adding INTACT, we designed experiments intended to assess the effectiveness of such a coaching tool within a comprehensive training program.

In Phase II, we will assess the value added by intelligent coaching concepts using quantitative performance measures associated with the DDD simulation environment. The measures will be model-driven, that is, they will refer back to the models of command and control coaching defined in Objective 1, linking models of command and control decision making to decision support concepts in order to predict the expected results of providing real-time coaching. In addition to model-based measures, we will select measures of effectiveness and measures of performance that were specifically designed in other ongoing research projects at Aptima to capture operationally relevant constructs of AWACS and command and control performance.

1.2. Domain Selection

As mentioned above, we selected the AWACS as our platform to develop INTACT. The rational for this decision was based on both the characteristics of the AWACS operational environment and existing domain knowledge within the project team. The AWACS is a rich command and control environment, on both an individual and team level, in its role within the Theater Air Control System (TACS) as a front line air battle manager. The AWACS is often referred to as an air traffic control (ATC) station in the sky, however, they are required to complete a wider ranger of tasks that their civilian ground-based ATC counterparts. Within the aircraft there are several sections, referred to as suites. These include the surveillance suite, self-defense suite, and the weapons suite. The weapons suite is in direct contact with the other aircraft in the airspace. Within that section there are at least 3 weapons directors (WD) and 1 senior director (SD). The senior director is the leader of the weapons section. The SD is responsible for configuring the weapons team based on the mission, reporting any problems to individuals outside of the section, taking over for overloaded, overwhelmed WDs, and performing other important missions within the mission (i.e. planning and revising the radio communication plan). Typically, the SD serves as a WD for several years prior to becoming an SD. The WDs are the air battle managers and conduct front line air traffic control. They direct friendly fighters to intercept targets, refuel with airborne tankers, return to base, etc.

A situational display console (SDC) serves as the system interfaces utilized by both the WD and SD. They display radar returns and other sensor data corresponding to aircraft that are within range (high enough, close enough, and of large enough signature, etc.). Those radar and sensor returns are "tagged up" with a symbol by operators in the surveillance suite. That symbol signifies the believed identity of the object responsible for the data. The broad classifications are, obviously, friend, enemy, and unknown. Within that classification, each radar return is assigned a track number. That track number is how the WDs communicate with each other.

The nature of the task of weapons direction is rather complex. There are tasks which are specific to each individual, yet the responsibility for weapons directing lies with the team. WDs must monitor all radar and sensor returns (hereafter called tracks) and make judgments regarding what to do about them. Here are several examples that provide a flavor for the weapons directing tasks.

- WDs must *monitor enemy tracks* to determine intent and predicted courses of action.
- WDs must fully understand the *rules of engagement*.

- WDs must orchestrate *fighter flow*.
- WDs must ensure the safety of *high value assets*
- WDs are responsible for conducting *search and rescue missions*..

The AWACS environment creates a high demand for *teamwork* (Elliott, Dalrymple, & Neville, 1997). In terms of teamwork issues within the WD team, there are several things to consider. The typical setup for the WDs is that each is responsible for a region, or sector.
SDs direct the WDs in several ways:

- The most common communication would be to remind the WD to do some "housekeeping" on his/her scope.
- SDs also monitor workload. They might direct one WD to help another or they may step in and do it themselves.

SDs communicate with the WDs several ways:

- Direct communication can occur as the SD might walk directly to the WD workstation.
- They may communicate via the nets using headsets.
- The SD can send alerts, alarms, and messages to the WD console via the computer system.
- Finally, the SD can send an arrow to the WD. This arrow points to a particular spot on the screen which the SD wants the WD to attend.

In addition to the rich team and individual command and control environment that AWACS provided, existing domain knowledge on the project team made the selection most viable. Aptima has received both Phase II and Phase III funding for an Air Force SBIR focusing on the use of the DDD as a team testbed for AWACS WD tasks and AWACS system and team-design issues, respectively. Each is described briefly below.

<u>A System to Enhance Team Decision Making Performance</u>: This program is developing a flexible team-in-the-loop simulation that can represent a variety of Air Force tactical Command and Control (C2) environments, embedding in it the characteristics of selected environments (e.g., AWACS, Common Operating Picture, Unmanned Aerial Vehicles), and testing alternative interventions designed to improve team performance such as shared information displays, group decision support systems, team coordination aids, and team embedded training. We have adapted a flexible synthetic team task, rooted in team theory and models, that is easily modified to represent teams in a multitude of environments. In Phase I, we demonstrated that the Distributed Dynamic Decision-making (DDD) simulation, with modest changes, could represent the team tasks of AWACS Weapons Directors. In Phase II, we are validating this representation and delivering a team performance tool package that includes measures, models, results, and specific guidelines, empirically tested in experiments, for effective team support interventions (shared displays, decision support, procedures, and training) to enhance team performance.

<u>Human-Centered Re-Engineering of AWACS Command and Control Teams</u>: This SBIR Phase III transitions the knowledge, research results, models, methods, and tools developed in Phase I and Phase II of the "System to Enhance Team Decision Making Performance" project to support the design of operational AWACS command and control teams for the AWACS Program Office at Hanscom AFB. There are four primary technical objectives for Phase III. One, we are

building a model of AWACS organizational structures using input from the operational community to define realistic and complex missions that serve as the input to our team modeling process. Two, we will optimize and redesign these organizational structures to meet future operational challenges, including new missions and ground-based AWACS operation, using our algorithm-based method of organizational design. Three, we will apply human-centered design principles to the evaluation of new technology to identify improvements to support enhanced performance and to develop integration strategies to avoid stovepiping. Four, we will assess the organizational impact of new technology insertion and create new architectures to ensure performance payoffs from technology investments.

1.3. Document Overview

The remainder of this document will elaborate on the introductory presentation of the INTACT development effort and the AWACS operational environment. The following are brief descriptions of the remaining sections of this document.

2. *Theory-Based Model of Command and Control Coaching*: Provides an overview of our theory-based model of command and control coaching entitled the Dynamic Coaching Protocol (DCP).
3. *AWACS Operational Needs Statement*: Describes the development of our operational needs statement that served as a key component in designing INTACT.
4. *Storyboard Description of Future Intact Implementation*: Illustrates in words and figures future directions for the development of INTACT, specifically focusing on intervention triggers, display methods, and a query function.
5. *INTACT Development*: Reviews the Phase I software development effort. Provides an overview of the DDD Simulator and the integration of the JavaCoach software within the DDD.
6. *INTACT Demonstration*: Contains a description of the INTACT system demonstration and reviews the operational focus of the demonstration.
7. *Phase II INTACT Software Development Plan*: Proposes our Phase II software efforts including specification development, software development, and software test and evaluation.
8. *Phase II Experimental Plan*: Describes an set of experiments to validate our theory-based model of coaching and the utility of the INTACT tool for training and operational support.
9. *Phase II Technical Objectives*: Presents an overview of our Phase II proposal and lists the anticipated benefits of these efforts
10. *Commercialization Plan*: Contains our assessment of potential post Phase II INTACT Applications and a plan to achieve these goals.

2. THEORY-BASED MODEL OF COMMAND AND CONTROL COACHING

A primary product of Phase I was our theory-based model of command and control coaching entitled the Dynamic Coaching Protocol (DCP). Appendix I contains a detailed description of the DCP and only a cursory review will be presented presently. The report outlines the theoretical approach behind an intellectual aide that serves to bridge instructional technology and decision support approaches in creating intelligent coaching systems. The approach is grounded in a technology that can monitor task and mediation properties in order to help provide levels of operational awareness that match operational demands. The objective is to build a framework, the DCP, which will lead to the development of adaptable software agents that can help guide expertise in military operations. An explicit metatheory will provide coaching principles that can be applied to a variety of operational contexts, whereas the triggering components of the coaching system will be scaled to the operational contexts at hand. The goal is to create a configurable coaching system that leverages context independent theoretical principles needed for a federated coaching software system.

A strong adaptable framework has been chosen in which to develop the cognitive engineering perspective illustrated in the proposal. First, a nomological network of ideas that forms the basis of an adaptable metatheory on human decision making and judgment is outlined. This network includes an overview of the metatheory of Probabilistic Functionalism, and its importance in specifying design options for optimized intellectual support is defined. Secondly, a corresponding adaptable methodology for implementing support through the use of an intelligent agent is outlined. A brief description of the DCP is presented in the remainder of this section.

The DCP is based on the consolidation of three areas of research: 1) judgment and decision making, 2) display engineering, and 3) situational awareness. The thematic features that unite these literatures can be summarized in the three global scientifically validated premises below.

1) There exists <u>variability</u> in task properties and these properties directly constrain expert decision-making. Before one asks any questions about the nature of information processing (i.e., what is going on inside the head of the decision maker), there must be a degree of clarity concerning the characteristics of the problem and the demands that are being placed upon the expert. This calls for a model of the decision task, or, more generally, a <u>methodology</u> directed at modeling the ecology in which the decision maker operates.

2) There exists <u>variability</u> associated with the mediation of the task system. The manner in which the decision environment is expressed will induce particular information processing responses in the decision maker.

3) There exists <u>variability</u> in the situational awareness of a decision maker. This variance is associated with task demands and mediation characteristics, as well as many individual difference variables that bear on the processing of complex data.

Using the three premises outlined above, we have created a coaching mechanism that is executed in a manner that maintains congruence between the task system, the mediation system, and the operational awareness system of the user. We believe there is a method to define a "band of congruence" that quantifies the degree to which the separate systems are aligned. Deviations from this alignment state will represent potential shortfalls in operator performance. For example, through implementation of a monitoring process for deviations in congruence, adaptable coaching interventions may be automatically invoked. Figure 3 illustrates this idea.

In Figure 3, a particular task configuration (T3 in Task State vector) will require a unique representation (M3 in Mediation State vector) that in turn activates or causes a given mode of cognition in the decision maker (S3 Cognitive Awareness State vector). This cognitive mode generates a particular course of action or decision (A3 Action Space vector). Fluctuations in alignment alter cognitive functioning (the dashed arrows emanating from the mediation vector) and cause poor performance (Error).

Two Coaches, one proactive and one reactive, are used to monitor and control aspects of the mediation State vector to maintain alignment. The proactive coach analyzes archival data and identifies error inducing situations through pattern matching and frequency analysis, and the reactive coach analyzes online data to evaluate decision quality and performance in real time.

The DCP model illustrates two intervention pathways: 1) proactive feedforward mechanism, and 2) a reactive feedback mechanism. The feedforward loop is viewed as important in communicating historical response information to the operator. Here, the adaptive component of the coach archives error profiles of an operator and then uses this information to trigger proactive alerts and operational awareness information to the operator. The feedback loop is triggered from an error event, which then activates the intervention toward an optimized awareness state.

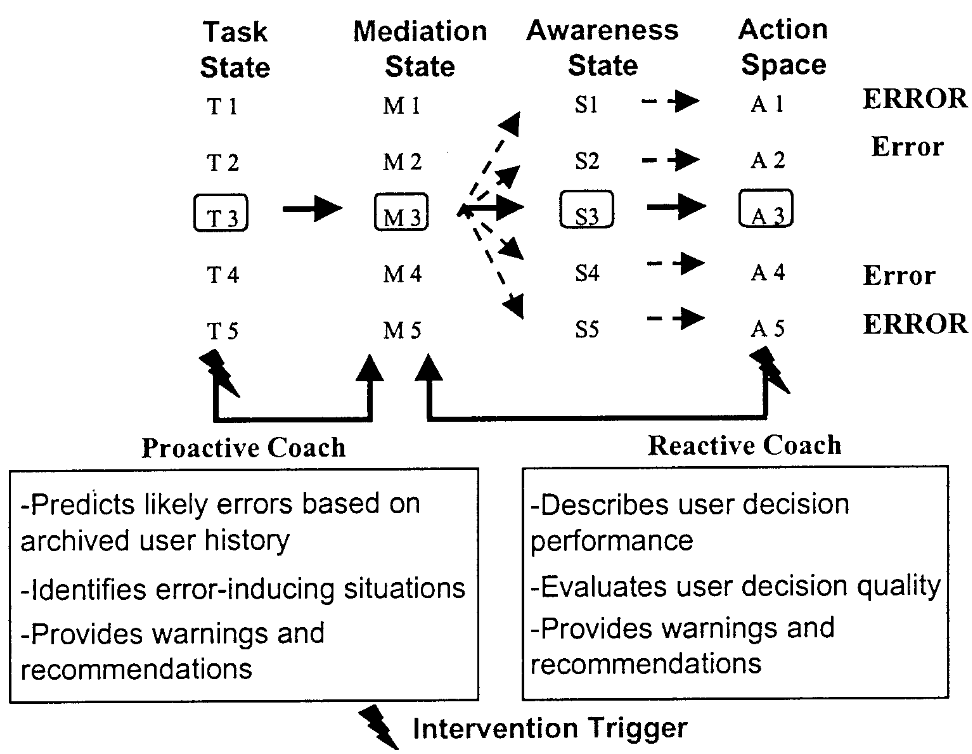


Figure 3. DCP Model Relating Task, Mediation, and Awareness States to Errors

3. AWACS OPERATIONAL NEEDS STATEMENT

The classical model of developing a training program is to conduct a training needs assessment to identify organizational weaknesses and define instructional objectives that serve as the basis for the entire training program (Goldstein, 1993). For example, an organization may find that their clerical staff is lacking advanced computer skills and that this deficiency does not allow them to gain full benefit from the company's computer systems. The likely outcome would be to develop a computer training program to address this shortcoming.

In developing a coaching tool, as described in our coaching model, we viewed coaching as something different from training. As a result, we felt that a traditional needs assessment would result in too narrow a specification of organizational needs for embedded support. Specifically, we wanted to identify operational tasks that were difficult and challenging to perform, as opposed to defining skills that the operators do not possess. The whole premise of the coach was that it would support performance by assisting the operators with tasks that they have already been trained to complete.

To identify the operational tasks that would drive the development of INTACT, we conducted a series of interviews with AWACS operators and subject matter experts to define an operational needs statement. An operational needs statement is intentionally different from the training needs statement in that it focuses on what is hard for operators to learn and perform, rather than what they do not know. The operational needs statement provides a mission-based justification for the proposed structure for the intelligent coach.

As stated, we based the design of INTACT to support the operational tasks of an AWACS weapons director (WD). The core of INTACT is a theory-based model of coaching that is applied to an operational domain. We collected domain knowledge about AWACS from a variety of sources during Phase I to develop an operational needs statement that highlighted areas of the WD's duty tasks that could be supported by an intelligent coaching tool like INTACT. In this sense, we were aiming to identify aspects of the WD's job that were either difficult to perform and/or are a common source of errors.

The operational data came from both the AWACS operational and research communities. The specific data collection activities are listed below briefly and in greater detail in the succeeding sections.

- Attended a training sessions at the Fighter Weapons School at Nellis, AFB and interviewed several instructors.
- Interviewed (via telephone) a member of the AWACS Wing Tactics Office at Tinker, AFB, and received and reviewed Wing Tactics Standards.
- Worked with representatives from the C3STARS Facility at Brooks AFB with respect to simulation-based training for the AWACS platform.

3.1. Fighter Weapons School

The USAF Weapons School, headquartered at Nellis AFB, teaches graduate-level instructor courses, which provide advanced training in weapons and tactics employment to officers of the combat air forces. The Weapons School provides graduate-level A-10, B-1, B-52, F-15C, F-15E, F-16C, HH-60, Command and Control Operations, Intelligence, and Space weapons instructor academic and flying courses to USAF Combat Air Forces (CAF). USAF Weapons School Command & Control Operations Division is designed to train officers to provide weapons and tactics expertise at the unit level. Students received instruction in GTACS, Regional Operational Control Center (ROCC), and AWACS operations. During our visit to the Weapons School, we observed a training session for one of the AWACS students and conducted interviews with three instructors.

3.1.1. Training Session

The training session is a live-fly exercise in which student pilots fly an air-to-air combat mission against instructor pilots. The AWACS command and control students control the fighter aircraft from a ground based control station. The session is a process of brief-execute-debrief. During the first phase the student presents to the instructor his or her plan for the upcoming mission. For the mission we observed, the student WD controlled a flight of four aircraft (four ship) against four enemy aircraft. Referred to as a "4v4" this exercise is intended to train the WD in the process of offensive counter air (OCA) maneuvers. The briefing required the student to step through the entire mission and served to highlight the operational demands placed on them during execution.

The briefing in many ways is a mission rehearsal. Using a white board, the student does a moment-by-moment enactment of the mission, including planed friendly actions and possible enemy responses. During the rehearsal, the student focused on what voice calls (messages to the pilots) to make at each stage, anticipating the communications they expect to hear, and anticipating the information needed to build the pilots SA (i.e., altitude and formation of enemy aircraft).

This process highlights the role of the WD as a provider of information to the aircraft they control. The reason that they plan ahead and anticipate pilot needs is that the operational load of reading and interpreting the data on their displays may distract from their ability to provide timely updates. From a coaching standpoint, understanding these communication needs and the mission characteristics that trigger them could serve as a very useful operational tool. A good example of this is the way targeting information is presented to the fighter. At ranges greater than twenty nautical miles, a WD will alert a fighter pilot of enemy aircraft with respect to a previously defined point in space or bull's eye. That is, the location of an enemy aircraft is given in terms of its relative position to the bull's eye. Within a specified range, the method of information passing switches to "BRAA" for bearing, range, altitude, and aspect. Rather than pass location information with respect to the bull's eye, the call is made with respect to the fighter's current location. A coach that could monitor communications could ensure that the right method of threat warning is used.

The other key element that was discussed during the pre-mission brief was preparing for the different phases of an intercept. Basically, there are different tasks associated with the different phases of an intercept (engagement with enemy in this instance, but it is a general term for a variety of missions). The student reviewed the different phases and the WD's main responsibility at each phase. A more detailed description of the phases of intercept are presented in the section describing the interview with a member of the AWACS Wing Tactics Office.

Presently, however, it is imperative to note that a coach could be a useful tool to support the operator in this area. One example of this was added to our Phase I INTACT demonstration in the form of providing safe passage to the fighters. The last phase of an intercept is referred to as egress and requires the operator to make a "green call" or provide the fighters with a flight path that ensures a safe return to base. In our demonstration we established a safe passage corridor that all fighters were required to exit the theater through. If an operator failed to vector an aircraft through this corridor, the coach would intervene. While we were able to incorporate the "green call" into our demonstration, the majority of the issues uncovered during the training session were too advanced for our Phase I effort. This should come as no surprise in that the Fighter Weapons School is an advanced program for highly skilled air battle managers.

3.1.2. Instructor Interviews

We conducted a series of informal interviews with three command and control instructors. The three, Captains Tony "Coach" Fournier, John "Uncle Buck" Iwanski, and Jeff "Sunny" Sundberg, are all AWACS senior directors and former graduates of the Weapons School program. Like the training session, much of the discussion with these instructors addressed concepts that were beyond the scope of our Phase I efforts. The interviews, however, provide insight into how to increase the utility of a Phase II INTACT. Without a doubt, the key to increasing utility is tied to the ability to provide coaching with respect to communication as suggested by this quote from one of the instructors, "Communication is the single most important issue for overall job performance."

The key area they would want a coach to support the training process is to help teach the WD to respond to the picture on the screen and the pilots needs. The WD must be ready with the right information at the right time and must provide this information in both proactive and reactive manner. In the former, the WD provides a "picture call" or an overview of the air battle theater to all the pilots on his or her radio frequency. In the latter, the WD must respond to specific requests for information from the pilot. Three main concerns here are "bogey dope" which is a request for information on a hostile track, "declaration" in which a fighter is asking AWACS to identify a track within the fighter's radar as friendly or hostile, and "clean" where a pilot has a blank radar picture and is requesting target information. The timeliness and accuracy of this data has a huge impact on overall mission performance, kills by enemy, and fratricide.

The issue of accuracy of communications is an interesting one. In addition to providing the right information, the WD must deliver it in the right format. There are specific standards that dictate the manner in which a WD should make a picture call and all other radio communications to fighter pilots. These standards are described in the 552d Air Control Wing Instruction 10–23 (see below) and are important for they keep the messages short and concise. An SD monitors the communications of the WDs in the weapons suite and it is quite common for a WD to receive feedback about the manner in which they present information. This type of coaching requires advanced voice recognition technology, however, a text-base mnemonic device could be explored.

In an attempt to address the issue of when to intervene, there were discussions of how an SD makes the decision to provide assistance to the WD. The instructors were asked what are the cues they are looking for that will indicate that a WD needs assistance. One instructor suggested that he is listening to what the WD is saying to get an idea of what their situational awareness is.

He was interested in if they are saying the expected communications, knowing what to say, and using the correct voice calls. In addition, he would monitor to hear if they are asking the right questions. For example, in the post merge phase, is the WD saying things and asking questions that indicate that he or she is trying to establish the location of the friendly aircraft.

The focus on communication is largely driven by the fact that during a flight, the SD is not able to look at all of the WDs' consoles. Given the ability to look over the shoulder of a WD, one instructor suggested that his main focus would be on the WD's screen settings and if he or she was properly using the tools available to perform the mission. Screen settings of interest include scale expansion and offset control (allow operator to re-center map display) because they provide a good idea of the operators focus and situation awareness. Likewise, how a WD is using the bearing and range controls can indicate his or her ability to provide relevant information to the fighters.

While these observations provide some directions for the intervention triggers for INTACT, a comment by one of the instructors provides a very interesting and important perspective. He said, "Nothing is too hard to teach or learn, the problem is load." His argument is that any given task that a WD must complete is not difficult to learn or perform, but executing the action at the right time when many things need to be done all at once is difficult. A WD must simultaneously deal with multiple aircraft, doing different tasks, with different resources and capabilities. As such, he argued that the bigger problem may be that the WD looses situational awareness and begins to "forget to do things." This corresponds nicely with the previous instructor's comments that he is monitoring the screen settings to assess situational awareness. From a coaching standpoint, INTACT should be developed to detect errors of omission in addition to errors of commission.

3.2. AWACS Wing Tactics

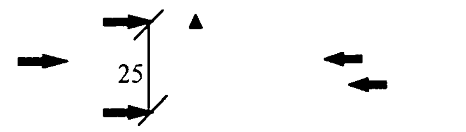
We conducted a series of telephone interviews with Master Sergeant Ted Hensley of the 552nd Wing Tactics, Tinker AFB. The role of Wing Tactics is to develop Command and Control (C2) standards for AWACS operations and to inform the Combat Air Forces (CAF) of these standards. That is, they develop the specific procedures that AWACS operators are expected to perform during a mission and inform the other airborne elements of the manner in which AWACS will perform their duties. These tasks are represented in the publication of the 552d Air Control Wing Instruction 10–23 (ACWI 10-23) Operations Employment Standards. An example, presented in Figure 4, of a standard from the ACWI 10-23 is how to provide a fighter with a description of the enemy air positions or picture call. The availability of these standards are viewed as potential input for the aforementioned mnemonic devices to support proper communications. A system could be developed to provide operators with communication cues based on operational events or in an on-demand fashion. Further exploration on how to best add communications coaching to INTACT will be a primary objective in Phase II.

Champagne labels: In a Champagne, the 2 groups closest to the fighters will be called **"lead"** groups (north lead group and south lead group), and the third group in a Champagne will be called the **trail group** (see example below).

"Chalice 3 group Vic,*azimuth* 25 *lead group*B/E 190/10 , estimate 20K.

From here provide the fill-in of how back the trail groups are: Chalice trail groups range split 12."

"Chalice 3 group Champagne, *azimuth* 25 *north lead* group B/E 270/10,estimate 20K.



From here provide the fill-in of how far back the trail group is: Chalice, *trail group* range split 12."

Note: When describing a **"4 group container"**, first describe the azimuth relationship, then the range.

Figure 4. Example of Picture Call Standard from ACWI 10–23

During the telephone interviews, we focused discussion on the most difficult aspects of a weapons director's job. What emerged was the notion that maintaining situational awareness of the five phases of an intercept is critical. As presented above in the Fighter Weapons School section (3.1), a tactical aircraft's mission is viewed in a five stage process that progresses through corresponding major events. Associated with each phase is a specific task or tasks that a WD is required to perform. These tasks, including specific communication formats, are defined within the ACWI 10-23. The specifics of the five phases may vary slightly depending on the mission type. Two examples from the 10-23 are presented here.

OCA/DCA Controller Standards (Air–to–Air):

1. **CAP/Detect Phase:** Detect (and ID) all contacts in the AOR. Provide situation brief using core information (controllers will make a picture call after friendly fighters turn hot in the cap, and when friendly fighters are half way through their cold leg of the cap). Ensure timely commit based on Desired Engagement Zone.

2. **Target/Sort:** When appropriate, label the group picture to aid targeting. QC group sort IAW briefed criteria or AFTTP 3–1. Provide amplifying data for specific groups to include number of contacts, ID, etc. Report previously unreported group maneuvers using FLANK/BEAM/DRAG. Maintain SA on untargeted groups as appropriate.

3. **Engaged Phase:** Update position of untargeted groups. Report group maneuvers as detected. Report inbound threats to the engagement area. Maintain SA on engagement locations/status. Provide assistance for flight rejoin if necessary.

4. **Merge Phase:** Maintain SA on merge locations and sanitize around them. Provide follow–on targeting and threat information, when applicable. Redirect other fighters to merge locations as necessary. Priority will be given to untargeted groups and defensive friendly fighters.

5. **Egress Phase:** Update untargeted group locations. Update friendly fighter locations using posits/status. Sanitize egress direction using "Threat" as appropriate. Identify fighters returning to the CAP location. Provide assistance for flight rejoin and Green calls.

Strike Controller Standards (Air–to–Ground):

1. **Marshal/Pre–Strike Phase:** Establish contact with friendly aircraft, obtain mission status. QC briefed IFF/SIF plan. Provide rendezvous assistance with pre–strike refueling. Provide situation update including airborne and ground threats.

2. **Ingress Phase:** Sanitize ingress route for air and new ground threats. Maintain positional SA on friendly fighters. Provide updates to threat information that will impact mission tactics or execution.

3. **Time Over Target Phase:** Sanitize target area for air and new ground threats. Report threat inbound to the target area. Maintain positional SA on friendly fighters. Echo "Magnum" calls as necessary.

4. **Egress Phase:** Sanitize egress routes for air and new ground threats. Identify/sanitize returning friendly fighters to assist with safe passage. Maintain positional SA on friendly fighters. Provide assistance for flight rejoin.

5. **Post–Strike Phase:** Regain contact with friendly fighters to obtain mission results and flight status. Provide assistance with post–strike refueling. Provide assistance for flight rejoin.

In addition to the above combat missions, the same basic five stage principle can be applied to other tasks, such as airborne refueling. In a refueling example, the five phases are 1) contact aircraft; 2) assign to tanker and direct; 3) time on tanker phase; 4) egress from tanker; 5) redirect to battle.

Discussions with Mst. Sgt. Hensley indicated that understanding what phase each aircraft is in, executing the correct task at the right time, and providing accurate information to the pilots is difficult for the WD to learn and to perform under conditions of high task saturation. A WD controls a number of aircraft and each may be at a different phase of the intercept at any given time. Therefore, the WD must maintain situational awareness for each aircraft on an individual basis. There is a tendency to focus on those aircraft that are in the merge or time on target phase of the mission. Yet, ignoring and omitting the latter phases is critical because of the increased likelihood of fratricide as fighters return to base. A friendly aircraft that is traveling towards other friendly aircraft may appear to be a hostile if not properly directed by AWACS.

As such, a coach can provide the operator with a mission management assistance that can provide information regarding the current phase of a selected aircraft. In this sense, the coach can be viewed as a mission information source, however, it can also provide reminders when action is needed. The coach can provide the operator with an alert when an aircraft needs to be tasked to perform the next stage of the intercept. As mentioned, we implemented such capability in our Phase I demonstration with respect to green calls or safe passage egress from the air battle theater.

3.3. C3STARS Facility

The Command, Control, and Communications Simulation, Training, and Research System (C3STARS) is an Air Force Research Laboratory (AFRL) asset housed at Armstrong Laboratory, Brooks AFB. The C3STARS is a high fidelity AWACS simulator. This facility has been used in a variety of research programs looking at operational performance on the AWACS. A specific focus has been on operational training, distributed mission training, and sustained operations. The C3STARS facility offers the opportunity to investigate complex decision making among interdependent team members within a dynamic and realistic setting. The crewstations and scenarios simulate the air defense mission of an AWACS platform. Realism is achieved through the functional representation of equipment and displays (see Figure 5), experienced personnel playing the role of simulation pilots, and the use of operational scenarios. We visited the facility, received a demonstration of the C3STARS capabilities, and meet with the research personnel.

Figure 5. C3STARS Simulated AWACS Crewstations

The objective of this trip was to obtain insight into the types of mistakes AWACS weapons directors are likely to make during a mission. We had a two day meeting with Mathieu Dalrymple. Mr. Dalrymple was an Officer in the United States Air Force from 1977 to 1989 where he gained extensive experience in Air Force C3 Systems, including serving as a Senior Director instructor on AWACS, Semi-Automatic Ground Environment (SAGE), and Control and Reporting Center (CRC). He serves as a research scientist and scenario developer for the C3STARS facility. In the former role he often serves as the SD during experiments with current AWACS WD as participants.

In addition to his AWACS knowledge, Mr. Dalrymple is familiar with the DDD and was able to present potential errors that would be relevant to a typical DDD scenario and those that are likely to be corrected by a SD who is looking over the shoulder of a WD. Of important note, these mistakes can best be described as beginner's (novice) errors and were selected to remain within the scope of a Phase I effort. Other more complex and difficult to identify mistakes were also discussed and will be presented as a key element of the Phase II proposal. That is, in Phase I the coach should be viewed as a tool to aid a novice in becoming a journeyman, while Phase II will address moving an operator from journeyman to expert.

We discussed the possible errors in terms of specific AWACS missions that would be relevant to our Phase I effort. In particular we discussed Offensive/Defensive Counter Air (OCA/DCA) and Airborne Refueling (AR). This is not an exhaustive list of AWACS missions, but rather is a representative sample that can be simulated and studied on the DDD simulator.

3.3.1 OCA/DCA

OCA/DCA refers to missions in which a WD is controlling a friendly fighter against a hostile aircraft in battle. In an offensive counter air scenario the friendly fighters are the aggressors and are attacking hostile aircraft. Alternatively, the friendly aircraft will be responding the a hostile aggression in the defensive counter air scenario. Despite these differences, the role of the WD is quite similar across both once the engagement has begun.

The first consideration for operator performance is the commitment of friendly assets to the battle. The term commitment and commit are used to describe the process of providing directional information to a friendly fighter to complete their assigned mission (i.e., location of an enemy aircraft). The process begins with a picture call from AWACS in which the location of hostile threats is presented to friendly forces. These picture calls are generally made in reference to a predetermined point in space or bull's eye. As the aircraft get closer to one another, AWACS will commit friendlies to specific hostiles and provide location information relative to the fighters current position. A WD must provide the right information and ensure that all friendlies are committed to hostile targets. Untargeted hostiles, as one can imagine, pose a significant operational threat.

The WD must also assess a number of variables about the fighters they control that will dictate the quality of the commit. Commit quality is affected by factors that can impact the likelihood of mission success. For example, a good commit must consider the amount of fuel that a fighter has (enough to complete mission and return to base) or ensuring that a fighter has sufficient number of the correct type of armaments to engage the enemy.

A second consideration is intercept geometry. Intercept geometry refers to the directions that a WD provides to a fighter pilot with respect to engaging the enemy. The selection of an intercept is primarily a rule based decision that considers the fighters armaments. For example, a fighter with AMRAM missiles is most effective if directed to the target on a nose-to-nose intercepts, whereas a fighter with infrared missiles would require a stern or rear intercept.

A third consideration is the issue of resource management. The WD is often referred to as an air battle manager and is directly responsible for ensuring that the air tasking order (ATO) is executed as specified by the battle commanders. However, in an air-to-air combat situation, enemy forces levels are dynamic and AWACS is responsible to target all hostile with matching force numbers. This goes beyond the issue of commitment stated above and deals with how well an operator employs the assets under his or her control, including requesting more assets to complete the mission as necessary. For example, a WD does not want to send two friendly fighters to engage six hostiles and may need to delay a mission to build up force numbers. Another significant consideration with respect to resource management is the issue of egress from the air battle theater. A WD is required to provide safe passage to the fighters. In general, a WD will provide the fighter with a cardinal direction away from follow-on threats, also known as a green call. In certain cases, a safe passage corridor may exist and a WD will direct the fighter to that airspace for a safe egress.

3.3.2. Airborne Refueling

The refueling of fighter aircraft is a common aspect of almost all air missions. A fighter will often take off from an military base, arrive in theater, and proceed directly to the refueling tanker. In addition, an aircraft may complete one phase of a mission, refuel, and then return to the battle. In both cases, the role of the AWACS WD is to direct the pilots to the tanker. There are several tasks that a WD must complete to ensure safe refueling procedure. First, because there are multiple aircraft in queue to receive fuel, the WD must stack the fighters in order, as specified by the ATO. Maintaining safe separation, in both distance and altitude is the primary area for error. Once the fighter is in position below tanker, the WD will hand off to the tanker at 1000 ft below the refuel cell. After refueling, the WD must pick-up the fighter within 5 miles of the tanker. These procedures are viewed as relatively easy tasks in and of themselves. However, there is difficulty associated with performing these task when multiple cells are in use.

3.3.3. Interventions

In addition to addressing the types of errors a WD is likely to make, there was discussion of the types of interventions that an SD might use if a problem was identified. It is believed that this information will be useful to shape the intervention techniques employed by INTACT in Phase II. As mentioned, it would be beneficial if the interventions were similar to those of an SD looking over the shoulder of the operator.

When errors occur or operational performance is deteriorating on the AWACS, the primary consideration for the SD is safety. A common response to the question of what happens if you make a mistake is that people die. In any live-fly situation, there is a fighter pilot with a limited view of the airspace that depends of AWACS to augment the picture. As such, it is no surprise that the SD's approach is to first tell the WD what to do, and then tell them why afterwards. The definition of tell them why is really a three level response. First, the WD should be told what went wrong. Second, a description of why it was wrong should be provided. Third, possible solutions, including various options and explanations, should be provided.

Clearly, a simulation based coach need not be constrained by a safety-based requirement for immediate disclosure of what to do. For training purposes, we believe that INTACT can substitute the "what to do" with a message suggesting that something is wrong. For example, referring to the OCA/DCA errors described above, Table 1 provides an example of the intervention for an incorrect commit due to low fuel in the fighter.

Table 1. Examples of Interventions for an Incorrect Commit

Level of Warning	**Response**
First warning	Invalid commit
Second warning	Fighter fuel level low
Third warning	Refuel Track 2376 prior to commit
Fourth warning	Vector Track 2376 to Tanker and use Track 3489 for commit

As can be seen, the objective was to let the operator know something was wrong, specifically that he or she is attempting an invalid commit. Based on this intervention, they could recognize the situation and revise the plan. Alternatively, additional levels of intervention could be obtained, either by operator selection or repeating the errors which will provide increasingly detailed options. Discussions with subject matter experts led us to believe that this process of increasing levels of specification would mimic the type of guidance from an SD in a training environment.

3.4. Summary of AWACS Operational Needs Statement

The operational needs statement for this project was developed through interviews and discussions with a variety of subject matter experts, both active duty and former AWACS operators and instructors, combined with background domain knowledge within the project team. The focus was on the AWACS WD and we found general agreement regarding the need for assistance and where to focus our design efforts.

There was agreement that the WD's task is inherently a multitask situation, but one that is well defined in terms of standards and procedures. As such, coaching should focus on providing assistance to the operator to ensure that the established methods are being employed. This can range from evaluating commit decisions to providing assistance in keeping track of the current phase of an intercept for each fighter. This is the focus of our Phase I INTACT prototype.

An additional point of agreement was that the role of communication is so significant within most mission tasks that ultimate mission performance is primarily a function of the WD's ability to pass information to the fighters. Based on this finding we have proposed that our Phase II effort focus on incorporating communications, via voice recognition software, into the DDD and using these inputs to trigger interventions.

4. STORYBOARD DESCRIPTION OF FUTURE INTACT IMPLEMENTATION

Our Phase I storyboarding effort focused on three distinct functional areas, intervention triggers, display methods, and a query function. Each is described in detail below.

4.1. Intervention triggers

In the course of discussing coaching with experienced AWACS Senior Directors and subject matter experts, they stated clearly that the decision on when to intervene and provide assistance to a weapons director is rather difficult. An SD has limited knowledge of a WD's actions. They often rely on the consequences of errors rather than the antecedents. This limited information set makes the decision more difficult. Yet even with additional data, the decision remains difficult and we recognize that the intervention triggers will be critical to developing a useful tool. Trigger development must be sensitive to the issues of misses and false alarms which are often pointed to as reasons for low operational acceptance of such systems.

Our approach will be to create a rule-base that defines the possible states of the system and the membership functions that define the relationships among the operational values. The Fuzzy Rule Matrix shown in Figure 6 graphically illustrates how the rule matrix is used to invoke display interventions in response to changes in operational awareness. Here, a fixed task state is paired with the optimal (proximity-based) display. The displays will range from tabular-numeric to closed-form graphic in order to represent the separable to integral continuum on which the Analytical-Intuitive distinctions are conceived (see Wickens and Carswell, 1995). In response to an error event or an historical marker which predicts an operator error event, the operator's responses are categorized within the rule matrix and evaluated for congruence properties. When departures from congruence are detected, a display intervention (change in display format) is invoked that <u>guides</u> the user toward target state.

For example, the a rule combination may look like the following:

- IF task category = low AND display category = low AND operational awareness mode = low
- THEN system = congruence (green cells (light gray) in Figure 6)
- IF task category = low AND display = low AND operational awareness mode = medium
- THEN system = Positive_small Deviation (red cell (dark gray) in Figure 6)

In this second condition, the coach's response is to provide the Positive_small Deviation state in order to invoke a minor change in the level of awareness and help move the user closer to a point of congruence.

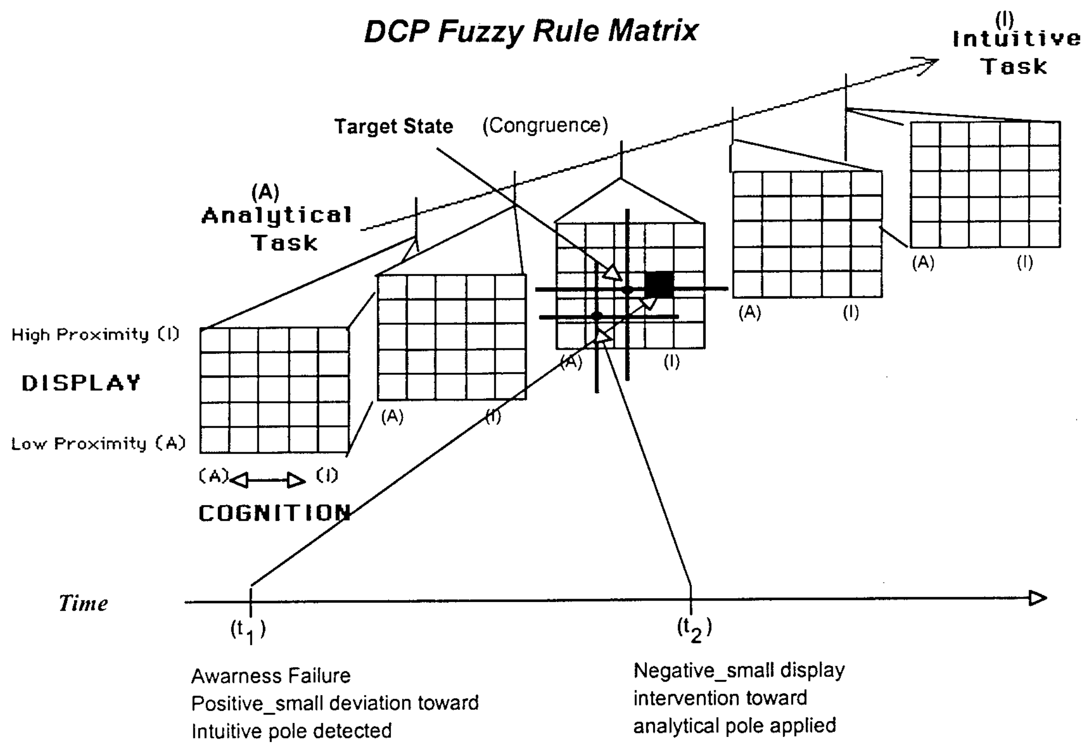


Figure 6. Dynamic Coaching Protocol rule matrix

4.2. Display Methods

Methods for displaying information in the coach will be selected based on the congruence principles of the DCP. The coaching mediation method will use both reactive and proactive display concepts. These two vehicles for delivering coaching information will provide the operational awareness responses to task specific errors and projected errors.

Figure 7 shows a storyboard for the coaching interface as it might be adapted to integrate into a DDD AWACS platform framework. Included in the interface is the primary display grid that presents two-dimensional positional information on military aircraft assets based on radar and sensor information. The Coach pull down menu is the interface to the DCP module. A number of options will be available including a switch that will activate the system to automatically monitor incoming communication and system data, as well as the responses of the WD to this information. A variety of algorithms are now under development that will measure parametric values of battlespace systems (fuel, distance to target/s, tanker locations, etc.). These algorithms will be the heart of situational awareness mechanisms that alert the WD to potential missed or neglected information important in a given tactical scenario. Within this mode, the WD can activate speech-based or text-based natural language probes of the coach in order to understand the rationale used by the coach in decision recommendations (see Query Function below).

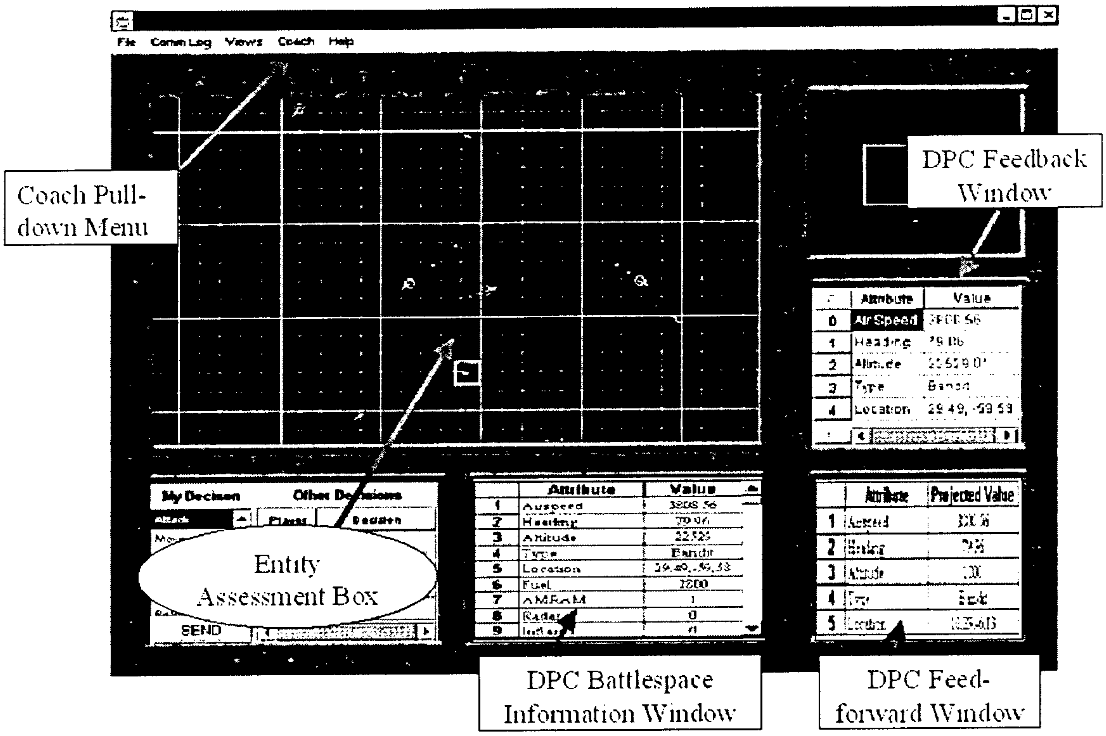


Figure 7. Coaching interface with feedback presented in tabular format

In Figure 7, an analytic-focus task state has been identified by the DCP system in response to an error linked to an uncommitted bandit nearing a high-value refueling asset. There are two immediate consequences of this detected error event. First, the DCP system highlights the bandit on the screen with an entity assessment box. Secondly, a DCP battlespace information window displays all of the critical attributes of the object that are known, such as heading, airspeed, radar type, etc. This window gives overall diagnostic information regarding the object. The DCP feedback window isolates the attributes that are associated with the error event in a table of values. These attributes provide specific information on where this bandit is relative to the refueling asset. The tabled values require an analytical assessment in order to precisely determine the bandit's location. The feedforward window extrapolates this information to provide a future situation-report on where the bandit will be in a given time period (configurable). The goal of the Coach is to induce in the operator an analytical assessment of this event in an effort to quantify what must be done in the immediate future. The need for attention focus in this example is due to the demand for a quantitative analysis on the bandit proximity to the refueling asset in terms of when the bandit will close with the asset. The Coach intervenes in a manner that produces a very narrow and elemental assessment of an object's precise position.

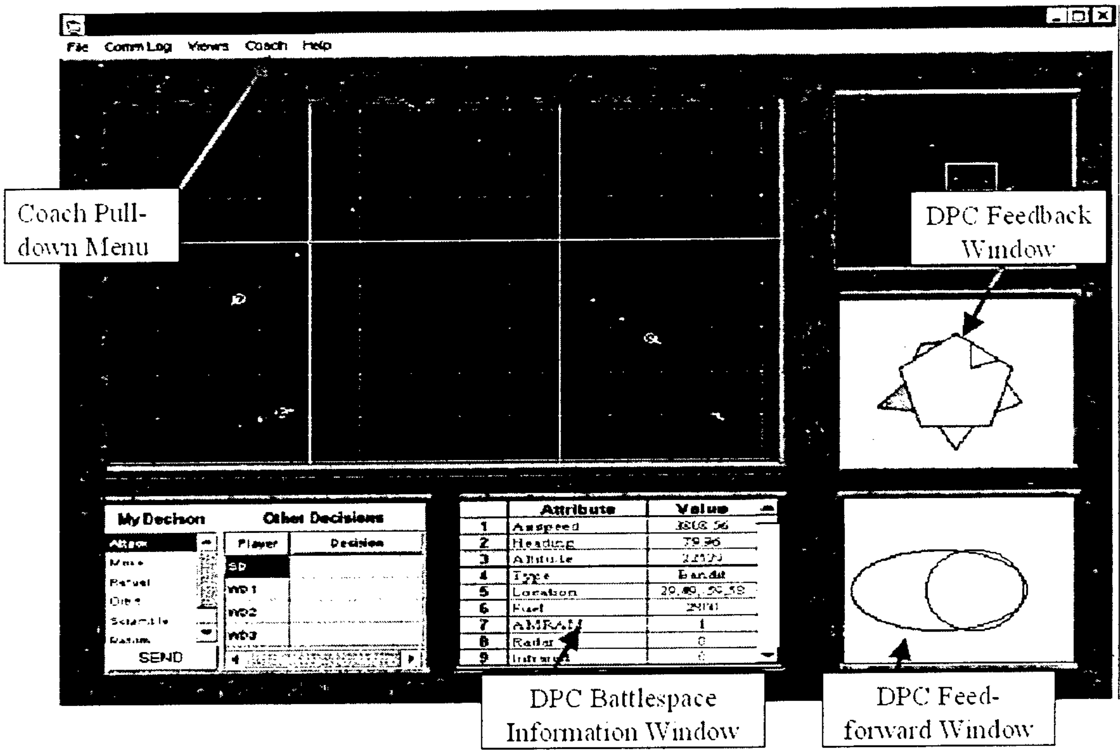


Figure 8. Coaching interface with feedback presented in graphical format

In the storyboard shown in Figure 8, an intuitive wide-focus task state has been identified by the DCP system in response to an error linked to a refueling alert. In this case three aspects of the DCP system have changed. First, there is no entity assessment box because this error source reflects a number of objects that must be evaluated in relation to one another and requires a broader level of operational awareness. In addition, the feedback window has become a polygon display that maps the fuel status of several aircraft to features of the polygon. Here a fleet of aircraft has been identified as needing fuel. The polygon provides perceptual information on the relative status of the individual aircraft. Since refueling will be completed on a need basis, the polygon display provides a very rapid way in which to assess the relative need of all the aircraft without quantifying each aircraft's fuel level. The feedforward display provides the future state of the refueling cell. Here, an animated display shows the refueling status of a tanker. As the display changes from a circle (open refueling status) to an ellipse (closing refueling status) the operator becomes aware of the timeline to route aircraft to the tanker before the number of aircraft at the tanker exceed the timely refueling capabilities of the cell. In this scenario, the goal is to provide a coaching intervention that facilities a very rapid form of processing multiple information elements.

4.3. Query Function for INTACT

One of the objectives of Phase II is to explore the explore the utility and need for adding a query function to the coach that would allow the user to enter into a dialogue with the coach to solve an operational problem, in real-time. Development of such a feature will require a formal development process. This process will begin with a utility analysis of the feature. We will present storyboard descriptions of the concept to potential users and assess the demand for such functionality. Assuming that there is support for such a feature, we will design the functionality based on the outcomes of user interviews. More specifically, we will focus on the critical incidents that would warrant a query function and build a tool that could provide useful tactical support to the operator in a time-critical fashion.

4.3.1. Example of INTACT Query Function

To clarify our approach, we give an example of how an intelligent coach might assist an AWACS WD (MacMillan et al., 1997). In this example, we assume that the model can be represented using semantic network procedures (see Simon, 1987). Our task is to develop a representational vocabulary for the model parameters and operands. The vocabulary chosen will, in principle, 1) provide a mechanism to formally justify an intelligent agent's value-based advice, 2) increase the level of awareness concerning how the advice is computed, and 3) facilitate iterative refining of model parameters.

A set of value based choices will be generated for a series of AWACS scenarios centered on the WD role. Refueling operations will be one of the exercises targeted for coaching support. Refueling is a good candidate because there are a number of factors that must be considered prior to and during refueling, particularly under combat conditions and with limited refueling assets.

The essential unit of information for the intelligent coach will be the rules that govern the operational procedures. We plan to map each operational heuristic recorded from expert users and training doctrine documentation directly into a rule and to encode a shared set of rules for performing supporting tasks. The rules for the agent will be grouped by function (e.g., query submission and information collection) and by task importance (as a function of tactical conditions under which tasks are being performed).

The rules will be categorized into functional classes. Each functional class is associated with a priority that determines which rule is activated whenever rules form multiple classes are satisfied simultaneously. The rules will include: 1) System control rules that create the internal model of the tactical environment and support actions requested by operators. 2) Periodic query submission and time-out handling rules that control the period querying of resources of the tactical environment. 3) Information collection and data reduction/explanation rules that collect target-system messages and update the system's internal model of the environment. Finally, some rules will expand information by supplying attributes with values that are implied by assessment of environmental conditions. The prioritization of rules will be used 1) to execute rule groups in a procedural fashion, and 2) to indicate the value of plans encoded by knowledge-based action rules. A second dimension in rule organization concerns the dynamic enabling and disabling of rules during expert-system execution, according to the current tactical state. For example, certain tactical states (radar detection of multiple hostile aircraft) will call for a different set of refueling procedures. This will require mapping ranges of severity of conditions to procedural modes.

Suppose a local Tanker is being used to service aircraft from a particular sector in a refueling operation. However, the tanker has experienced mechanical problems that prevent it from refueling additional aircraft at this moment in time. The coaching tool may be called upon explain and refine its choice among the following actions for refueling:

- DELAY: Delay refueling operations and maintain combat aircraft in area of responsibility.
- WAIT: Send aircraft component to local tanker to wait on local tanker.
- ALTERNATE: Send aircraft to an alternate tanker in a different and more remote sector.
- ALTERNATE PRIORITY: Send the most needy aircraft to remote Tanker.
- STAGGER: Send aircraft to both the local (disabled) and alternate (distant).
- HVA CALL: Request that a new tanker be sent to the WD's location.
- LANDING: Pass aircraft to the Check-in WD to land and refuel on the ground.

The structure of a possible interaction between a WD and the coaching tool might be as follows. A request for a choice will be made by the WD. The agent will 1) survey and collect immediate information on operational status of mission, 2) evaluate options on the basis of the a defined valuation model, 3) integrate the choices with refueling objectives, 4) make recommendation to the WD. Table 2 shows a hypothetical dialog between a WD and a proposed INTACT query function. The intelligent coach can be viewed as a storehouse of intuitive and evolving heuristics for managing resources in a tactical environment, which reflects the goals of the ATO (air tasking order) as well as fragmentary orders, and a number of tactical state values. The inclusion of an explicit model of value such as a multi-attribute utility model provides a normative basis for organizing the agent's numerous operational heuristics and for justifying those heuristics to WDs.

Table 2. Hypothetical Dialog Between WD and INTACT Query Function

WD Question	Intelligent Coaching Agent Response
Of the six alternatives above, which is the best?	ALTERNATE PRIORITY. This choice is best with regard to overall refueling effectiveness given the state of operations at this time.
How long will this take?	• 11 minutes for aircraft to remote tanker. • 4 minutes per four aircraft refueling (refueled 4 per time) • 14 minutes for aircraft return fueled • Total time for 30% strike package = 67 minutes
Which choice would be similar to Alternate Priority with regard to overall refueling effectiveness?	Two choices- WAIT PRIORITY and WAIT. The fact the mechanical problem is minor and that the local tanker will be open in a short amount of time makes both of these options appealing. The superiority of ALTERNATE PRIORITY choice is based on the consequence of a delayed mechanical problem with the local tanker and the unknown status of enemy activity within the AOR.
What factors are important in the determination of refueling effectiveness?	Speed in Meeting the demands for combat readiness.
What choices are best for immediately ensuring readiness for combat?	DELAY refueling, WAIT PRIORITY, and WAIT. All three of these choices, rank order maximize the capability for threat response because the aircraft remain with the strike group.
Give me a summary-level comparison of WAIT and WAIT PRIORITY.	WAIT includes sending components of the strike package to queue-up at the local tanker. WAIT PRIORITY includes sending only certain aircraft that have been evaluated on the basis of need and function. WAIT PRIORITY is a more conservative refueling option because more aircraft remain combat ready.
What are the costs of the WAIT PRIORITY alternative?	The WAIT PRIORITY places the aircraft on a cyclic refueling schedule that must be maintained until all aircraft can be refueled at the same time. During a WAIT PRIORITY schedule, there will always be some aircraft that need refueling and some that do not. This fact limits both offense and defensive capabilities because aircraft cannot coordinate activities in mass (simultaneously).
DELAY	You have selected DELAY as the alternative. ALTERNATE PRIORITY refueling is favorable to DELAY. If you <u>disagree</u> with this statement then you must <u>disagree</u> with at least one of the statements that supports it: Which of the following statements are you least comfortable with? 1. ALTERNATE PRIORITY is based on the expectation that while the problem th is estimated to be minor, the possibility that it cannot be fixed in a timely fashion m considered given our current knowledge of the tanker. 2. Refueling is critical to mission.
2	You selected "refueling is critical to mission" This is a primitive statement that you can modify directly. Do you want to change this?
Yes	With no other modifications, you must lower the value of "refueling is of critical importance" in or to prefer DELAY. The current value on a 10 point scale for refueling is 9 (extremely important). What is the correct value?
7	OK. DELAY is now the optimal choice.

5. INTACT DEVELOPMENT

5.1. Distributed Dynamic Decision-making (DDD) simulator

The simulator that we identified as appropriate for INTACT development is the Distributed Dynamic Decisionmaking (DDD) simulator. The DDD simulator has been a major component of a team research program that has been underway for almost 15 years. The DDD simulator, co-developed by David Kleinman and Daniel Serfaty, is a unique software tool set and computer system that originated in 1984 to study issues of distributed situation assessment and resource allocation in a dynamic team environment.

The DDD is a distributed client server simulation that provides a flexible framework in which to study team performance. DDD simulations involve team decision making about complex situations based on information and resources provided by various team members (Serfaty & Kleinman, 1985; Kleinman & Serfaty, 1989). In a typical DDD scenario, a team of decision makers must make coordinated decisions based on uncertain, ambiguous, and sometimes decentralized information. Each team member has only a portion of the needed information and/or resources to accomplish the team task The task may be easily configured for teams of up to seven members, networked together to form a variety of organizational structures.

In order to create the task environment, the DDD simulator generates dynamic scenarios of external ("world") events presented to the decision makers through a set of graphical and alphanumeric displays. Figure 9 shows a configuration for a team of four decision makers—one leader and three subordinates—with typical DDD team decision-making tasks in such a configuration.

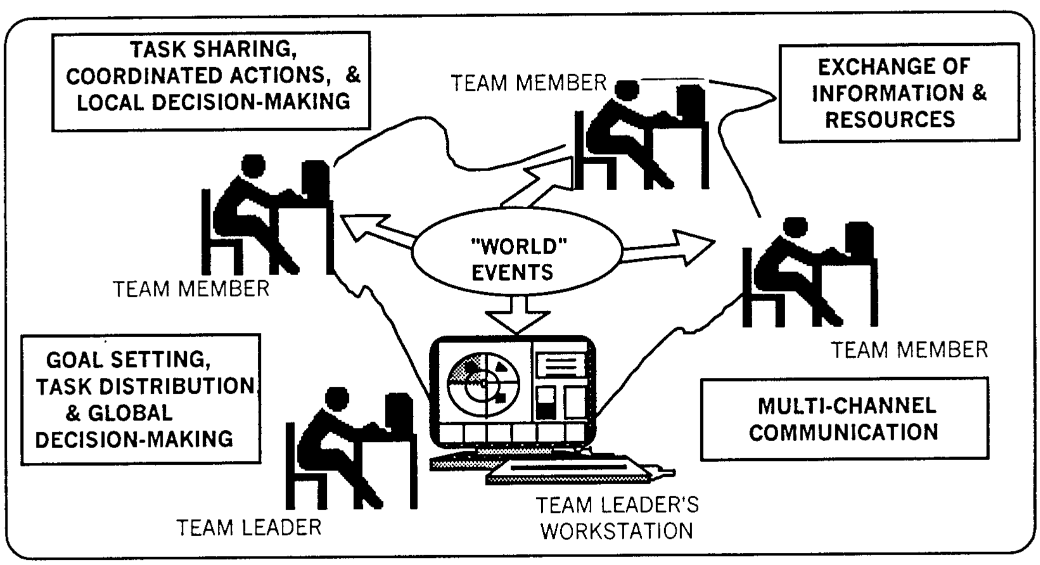


Figure 9. Typical DDD Configuration and Team Decision-Making Tasks

A team task such as the one illustrated in Figure 9 re-creates many of the cognitive demands associated with team decision making and cooperative work in a command and control environment. The DDD simulation task is designed with a flexible structure that can be manipulated to vary a number of different elements of task complexity (e.g., risk, uncertainty, time-pressure, information distribution, communication structure.) The DDD software provides real-time control and on-line data collection during training experiments, an interactive display/interface media, and a computerized intra-team communication sub-system. The DDD simulator is capable of recording over 100 dependent variables that fall into three major categories: performance measures, individual and team decision processes measures, and communication/coordination measures.

The DDD simulator has often been used to study military decisions in a distributed tactical environment. It has also been reconfigured to represent manufacturing and production scheduling problems (Wang, Luh, Serfaty, & Kleinman, 1991) and decentralized medical diagnosis in teams (Pete, Pattipati, & Rossano, 1991). The current generation of the DDD—DDD-III—is a distributed real-time simulation environment implementing a complex synthetic team task that includes many of the behaviors at the core of almost any command and control team task: assessing the situation, planning response actions, gathering information, sharing information, allocating resources to accomplish tasks, coordinating actions, and sharing or transferring resources. Figure 10 presents the AWACS Configuration of the DDD.

Measures that can be collected in DDD-based simulation sessions include individual and team performance, as indicated by such factors as the number of accurate identifications made, the number of targets successfully intercepted, or the number of tasks successfully completed. It is also possible to measure process variables that provide insight into decision strategies and team coordination, such as the speed with which different tasks are processed, the existence of uneven task loading across team members, the transfer of information among team members, and the transfer of resources among team members. Communication measures can also be collected.

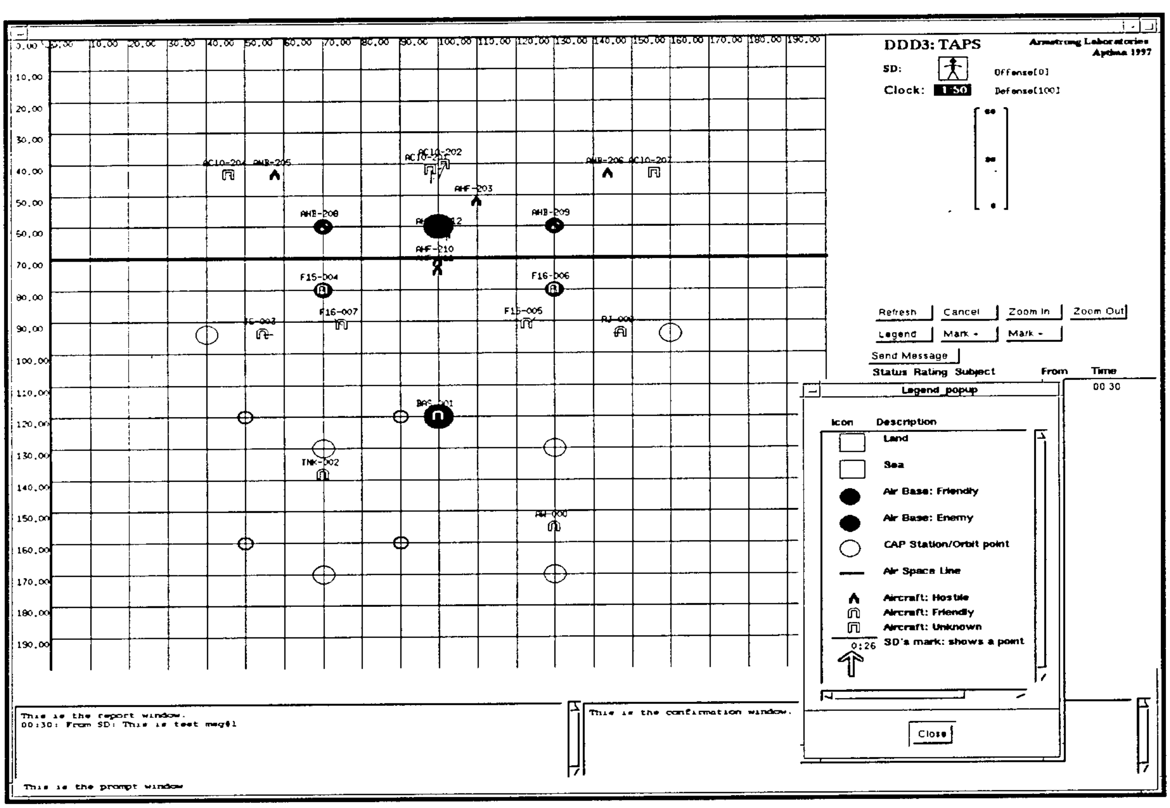


Figure 10. AWACS Configuration of the DDD

The results of DDD-based experiments have been demonstrated to carry over into real-world environments. In the TADMUS program, initiated in the wake of the Vincennes incident, the DDD was used to identify coordination strategies that are effective for teams under stress. The DDD simulated the tasks of the anti-air warfare team in the Combat Information Center (CIC) on Aegis-equipped ships (like the Vincennes). Insights from DDD-based team experiments with Navy CIC officers led to the development of *Team Adaptation and Coordination Training (TACT)* (Serfaty & Entin, 1995; Serfaty, Entin, & Johnston, 1998). The standard training provided to Aegis CIC teams now includes principles from TACT.

In order to test intelligent coaching concepts, we will draw on the flexibility of the DDD and its extensive research history to select a range of configurations and scenarios that create challenging situations for command and control decision making. The DDD has long been used to abstract military environments in which tasks/tracks must be identified/classified and, if hostile, processed/attacked. A "task" has an identity and a set of attributes that in turn determine what resource requirements (or capabilities) are needed for its successful prosecution. Friendly platforms (assets) provide the WD team with a distributed set of available resources that must be allocated to the task(s) in such a manner that the resources allocated meet or exceed the resources required.

The basic problem that a WD faces for a <u>single</u> task is: (1) identify the task's class (if not already done so by the surveillance team), (2) determine the task's resource requirements (i.e., which combinations of weapons are needed), (3) decide whether to process the task and when, (4) position/move/combine selected platform(s) to best meet the task's resource requirements, and finally, (5) process the task.

The challenge is that these activities must be done in a dynamic, multi-task, multi-person, distributed environment. It is this environment of distributed resource allocation under uncertainty that the DDD captures well in a low-fidelity, but highly flexible, synthetic task simulation.

5.2. DDD and Coach Integration

In Phase I we implemented a prototype for the INTACT coaching tool, using the DDD simulation as a testbed, and implementing the coaching capabilities with a Java based software program, referred to as JavaCoach. As can be seen in Figure 11, the JavaCoach is a single windows element that provides text-based information to the user.

The DDD has two responsibilities with respect to the JavaCoach. The first is to set up and maintain a network connection to the JavaCoach program. The second is to communicate with it in real time to inform it of events. We defined a communication protocol between the coaching tool (Java) and the DDD (C++) using a TCP/IP socket. This interchange is based on an XML text protocol which allows us to take advantage of emerging web technologies to enhance performance in future iterations. In order to shorten the development process and to leverage current DDD code structures, we developed the coaching trigger within the DDD and not within the Java coach. As such, the only messages passed from the DDD to the coach software are triggers.

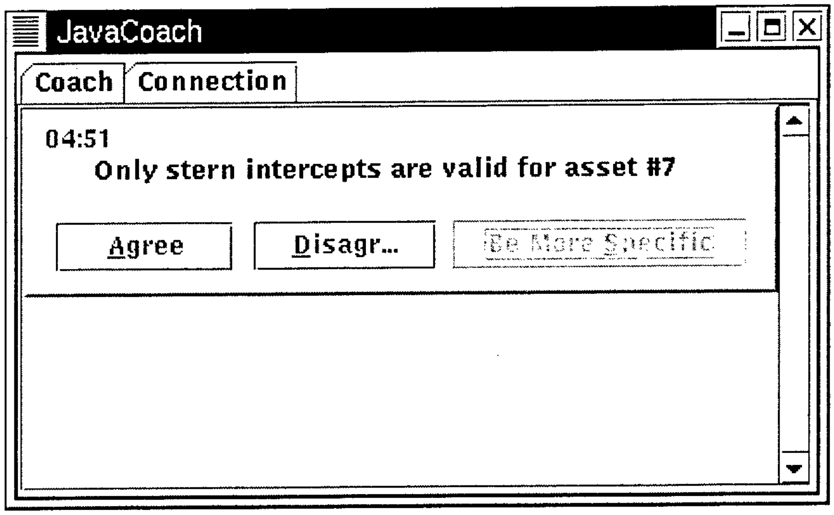


Figure 11. JavaCoach Interface

In Phase II the trigger mechanism will be incorporated into the coaching software. This integrated software package will be the realization of INTACT. Our intention is to design this software to be High Level Architecture (HLA) compliant which will allow for very efficient communication between the DDD and INTACT. We will send both operational state and operator action messages from the DDD to INTACT which will use the information to devise proper coaching strategies. There are two main advantages of this approach. First, because INTACT will be receiving output data directly, it can build profiles of the users based on their actions and the situations that caused them. In this sense, INTACT will have the ability to build accurate models of the user, update them over time, and therefore individually tailor the pedagogy to each user. The second advantage of the HLA-based design is that INTACT will act as an HLA object capable of receiving data from other HLA compliant simulators. This flexibility will facilitate the transition to alternative platforms as we execute our transition plan. For example, if we decide that the Uninhabited Combat Air Vehicles (UCAV) operations center is a viable platform, we would be able to use the core of INTACT, including the coaching pedagogy, to communicate with a HLA-compliant UCAV system and greatly reduce development costs.

Additional detail about the software aspects of the DDD and the JavaCoach integration are provided in Appendix 2, *DDD-JavaCoach Software Users Guide.*

6. INTACT DEMONSTRATION

6.1. Operational Focus of Demonstration

In the Phase I prototype version of INTACT, the focus was to provide real-time coaching for a limited set of DDD tasks. Two general tasks were selected, pursuit of hostile aircraft target and providing safe passage to friendly assets to exit the area of engagement. For the pursuit of a target, the JavaCoach is informed if the asset is too low on fuel for the mission, if the target is not in the proper zone for engagement, or if an inappropriate intercept geometry is selected for the pursuit. From an operational standpoint, this addresses the quality of a commit, correspondence to the rules of engagement (ROE), and resource management, respectively. For leaving the area of engagement, the JavaCoach is informed if the planned path of the asset will take it through a no-fly zone for exiting (Green Call), and it will again be informed when the asset actually enters the zone. Trigger events to initiate an intervention by the JavaCoach for both classes are described in the following sub-sections.

6.1.1. Pursue Trigger Events

When the user selects "Pursue" from the asset menu, the DDD checks if there is enough fuel for the asset to travel to the targeted task and back to a refueler. If not, then the Coach is sent a "Commit Fuel Check" event. Next the DDD checks the location of the task. If it is within the enemy airspace, then a "Commit Rules of Engagement (ROE)" event is sent to the Coach. If the enemy aircraft enters the friendly airspace and is not yet committed, then a "Commit Rules of Engagement" event is sent, with a different error code to indicate that the targeting is needed.

As part of the "Pursue" action, the DDD pops up a dialog box from which the user must select an Intercept Geometry (ICG), or choose to cancel the attack. If the user selects a specific ICG, the DDD checks if it is appropriate for the weapons that the asset is carrying. If not, then it sends a "Commit ICG Check" event to the Coach, along with an error code indicating which of the two types of ICG errors were made. (One type is to select "Stern" or "Stern-Convergence" for radar weapons; the other is to select "Cutoff" for infrared weapons.)

6.1.2. Egress Trigger Events

To signal that an asset is ready to leave the area of engagement, the user selects "Exit Theater" and then clicks on the map to where the asset should travel next. A safe passage corridor was established with no-fly zones surrounding it. The proper way to exit the airspace of engagement is to click at the endpoint of the first leg of a path that would take the asset back to the safe passage corridor without passing through the designated no-fly zone. After that endpoint is reached, the user should select "Move" or "Exit Theater" once again, and click to the endpoint of the next leg on the path, directing the asset back into friendly territory.

If, after selecting "Exit Theater", the user defines a path that would cause the asset to travel into the no-fly airspace, then a "Commit Egress Check Event" is sent to the Coach, along with an error code indicating it is only a potential error. The asset has not yet entered the airspace; the user can change the path before it does so. When the asset enters the no-fly airspace after "Exit Theater" has been selected, and before it has returned to friendly territory, a "Commit Egress

Check Event" is sent to the Coach along with an error code indicating that this was an actual entry into a no-fly airspace (i.e. failed Green Call.)

6.2. Description of INTACT Demonstration

Figure 12 shows a view of the INTACT version of the DDD with the JavaCoach activated in the upper right hand corner of the screen. The various shading indicated the different airspace within the air battle theater. The yellow at the top of the map indicates enemy territory where, according to the ROE, enemies may not be targeted. The orange represented the active engagement zone and the safe passage corridor. The red area represents the egress no-fly zone. Finally, the green zone at the bottom of the map represents friendly airspace.

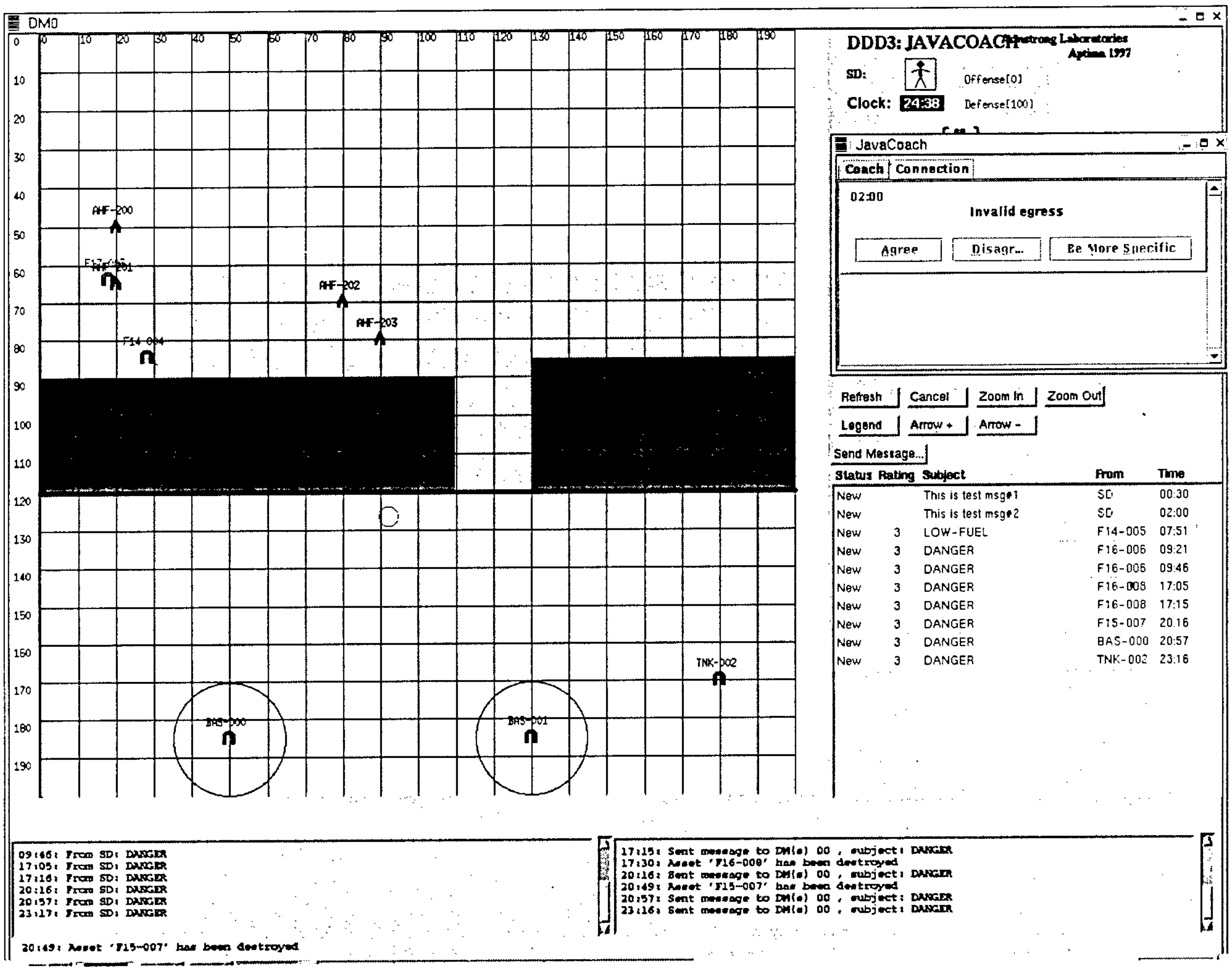


Figure 12. View of DDD Integrated with JavaCoach

Table 3 summarizes the "script" of the demonstration. The left-hand column indicates a Friendly asset. If the next column lists a hostile asset, then the action taken was to choose to pursue the hostile with the asset. The "Type of Pursue" indicates the Intercept Geometry chosen. When the events relating to intercept geometry were not being demonstrated, the user simply selected the general "Pursuit" choice from the Intercept Geometry menu. The other choices available were "Cut-off", "Stern", and "Stern-Convergence". Either Stern or Stern-Convergence were considered inappropriate geometries for an asset (such as the F14's in this scenario) that contain only radar-based weapons. Cut-off geometry was considered inappropriate for an asset (such as the 15's in this scenario) containing only infrared weapons.

Table 3. INTACT Demonstration Script

Friendly	Hostile	Type of Pursue	INTACT Intervention
F17-003	AC10-201	Pursuit	None
F14-005	AC10-202	Pursuit	Invalid Pursuit - Fuel
F16-006	AC10-202	Pursuit	None
F14-005	AC10-203	Pursuit	Invalid Pursuit - Fuel
F16-008	AC10-206 AC10-205	Pursuit Pursuit	ROE Violation None
F15-007	AC10-204	Cut off	Invalid Intercept - Weapon mismatch
F14-004	AHF-200	Stern	Multiple interventions ROE Violation Invalid Intercept - Weapon mismatch
F17-003	***	Exit Theater	Egress warning
F17-004	***	Exit Theater	Airspace violation

While the event in the demonstration were scripted, the interventions were not. That is, we planned to make specific errors to demonstrate the capabilities of the INTACT system, but did not provide this information to the JavaCoach a priori. In the event that an error occurred, the JavaCoach would present an intervention in the window. The operator was able to agree or disagree with the assessment or ask the coach to be more specific (see Figure 11). Agreement indicates that the operator recognizes the mistake and acknowledges that appropriate action will be taken. The disagree option allows the operator to have the JavaCoach ignore a particular action. In this sense, the operator is indicating that they have a reason for committing the violating action and do not want additional interventions with respect to this track. Finally, using the be more specific option provides more detailed descriptions of the error and possible corrective actions. The increasing level of specificity can, as stated, be obtained on-demand by the operator. In addition, the JavaCoach is programmed to provide more detail as errors persist or if similar errors are made repeatedly. Table 4 provides all the errors and JavaCoach interventions used in the Phase I demonstration.

Table 4. List of Errors and Interventions for INTACT Phase I Demonstration

Type of Error	JavaCoach Intervention
Low fuel	• Invalid commit • Fighter low on fuel • Refuel Asset #5 before commit • Direct asset #5 to base 1 for refueling
Rule of engagement	• ROE violation • Asset #8 violating ROE • Asset #8 should not target track #206 • Asset #8's target within enemy zone
Egress – current path will intersect no-fly zone	• Invalid egress • Possible egress without safe passage • Invalid green call for asset #8 • Vector asset #8 to safe passage corridor
Egress – entering no-fly zone	• Invalid egress • Fight without safe passage • Provide green call asset #8 • Vector asset #8 to safe passage corridor
Weapons-intercept mismatch	• Invalid intercept • Fighter weapons do not support intercept • Select alternative intercept for asset #7 • Only stern intercepts are valid for asset #7
Weapons-intercept mismatch	• Invalid intercept • Fighter weapons do not support intercept • Select alternative intercept for asset #4 • Only cut-off intercepts are valid for asset #4

7. PHASE II INTACT SOFTWARE DEVELOPMENT PLAN

The Phase II INTACT development effort is comprised of three subtasks, Software specification development, Software development, and Software test and evaluation, each of which is described in detail below.

7.1. Software Specification Development

Based on our experience in Phase I and additional knowledge engineering effort we plan to conduct in Phase II, we will author a document set detailing tool and end-user requirements and system specifications. Although we based our initial Phase I proof of concept on Java, in Phase II we will review the requirements and specifications, system descriptions, user guides, etc. of other software packages to identify any additional features or capabilities that would make our product more robust. We will be especially interested in assessing the relationship between the various software packages and developing HLA compliant objects.

The hardware and operating system will be selected based on customer requirements balanced against software development considerations to mitigate risk. The DDD, for example, runs on IBM PC or compatibles running Linux and on Sun and Silicon Graphics machines running UNIX. While we have demonstrated in Phase I that we can design coaching software to run on the Linux platform, we must assess the ability to operate our software as a platform independent object. This requirement originally drove the decision to develop a Java-based coach, but alternative software options exist and will be evaluated. One promising area that we are very interested in evaluating is designing INTACT to take advantage of WWW and database technology in order to be capable of interfacing with a variety of wargame simulations.

Based on the detailed specification of our embedded coaching tool requirements, areas of enhancement for INTACT will be identified. Some expected areas of coaching enhancements based on our Phase I experience are as follows:

- Voice recognition and voice generation capabilities to capture the communication and coordination aspects of command and control operations.
- Innovative display technologies to enhance the capability to provide users with coaching interventions in an unambiguous and easy to understand manner.
- Development of a performance summary output that can support training programs (i.e., automated "hot wash").
- Personalization methods that will allow the coach to custom tailor pedagogy to each user. For example, user profiles can be stored on a floppy disk and loaded into the INTACT system prior to each mission.

A set of requirement specification documents will be developed. An example of these requirement specifications is presented in Table 5. The first document will be a functional requirements specification detailing, from an end-user standpoint the functionality that will need to be included within INTACT. Then, software development specifications will be developed. We will develop a user interface specification for the INTACT detailing the look and feel, data required, and the dynamic operation of the tool. The user interface specification will define methods for the display of coaching interventions, training modules, and will describe the manner in which the user will input data into the system (if necessary).

Table 5: Sample top level INTACT Requirements Specification

Top Level requirements for Graphical Editor front-end
• GUI shall be driven by end-user characteristics, e.g., familiar metaphors.
• GUI shall guide user via intuitive display design, feedback, help.
• GUI functionality shall be bounded by the problem domain.
• From user data entry, system shall fill in simulation logic.

Some key considerations for requirements development are summarized below.

1) We will use for requirements development the IEEE Recommended Practice for Software Requirements Specification (IEEE Std 830-1993) as a guideline. The requirements defined will be considered our baseline. Project Management Plan will follow IEEE Standard for Software Project Management Plans (IEEE Std 1058.1-1987).

2) Aptima and University of Georgia will mutually define requirements and external interfaces. After the requirements development phase, software development of INTACT will become an external dependency to DDD development and the two software efforts should be kept independent of one another to the greatest extent possible to ensure generalizability of the final product.

3) Within the Software Project Management Plan, we will need to address the issues associated with this relationship and interdependencies. Unit/system testing and Verification & Validation will need to be included within the plan including specification of milestones and deliveries. The Project Management Plan will include discussion and mitigation plan for risks. For example, 1) handling the inherent complexity in developing new code and integrating existing code; 2) setting and managing appropriate user expectations; and 3) interfacing and interdependency between Aptima and University of Georgia parallel software development efforts.

4) We will explore implementation of an HLA Interface as specified by IEEE P1516/D1 Draft Standard [for] Modeling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules Specification. This standard or rules document describes the general principles defining a HLA by delineating ten basic rules which apply to HLA federations and federates. We will address each rule and document how our development process accommodated each one. Finally, we will conduct compliance testing as need and as defined by the Defense Modeling and Simulation Office (DMSO)

7.2. Software Development

All major features of the proposed design, including those implemented in Phase I and those defined in the requirements will be implemented. The major features that the software will support are: **real-time coaching interventions**; creating and maintaining a **dynamic models of the operator's actions and the tactical situation**; an **intervention trigger component** that does continuous assessment of both the operator and environment models and makes determinations of the need to intervene, and a **query function** for the operator to interact with the coach. We plan to explore the possibility that the coaching software will be able to communicate with the DDD through a high level architecture (HLA) and thereby be interoperable with other HLA compliant simulation tools.

7.2.1. Interface Prototyping and Implementation

A major component of the software development process will be to design and develop testable product prototypes prior to full-scale development. These prototypes will allow us to assess the functionality and usability of design alternatives before committing resources to full-scale development. Listed below are processes we will employ during the prototyping and implementation phase.

- **GUI Design Guidelines** - Develop and produce guidelines aimed at establishing quality and consistency (e.g., Human Computer Dialogue, Use of Symbols and Color, Screen Layouts).
- **GUI Development Environments** – Create a development environment which includes shared GUI widgets/modules, a database of common error/feedback messages, a set of GUI design standards.
- **GUI design and rapid prototyping** – Develop prototype displays based on human engineering analysis results and GUI design guidelines which can be evaluated by users to assess the extent to which the proposed GUIs support user tasks in an effective manner. Refine the GUI based on user feedback.

7.2.2. Software Usability

The goal of usability testing is to assess the ease-of-use of the software's capabilities and features. We will assess INTACT on the extent to which the tool is able to provide information to the operator. The emphasis here is not so much on how well the system performs as on its ease of use. The focus is on understanding how well the tool modules enable the users to learn and perform realistic tasks. Usability testing is not a point-in-time event. Rather, we will conduct these types of tests throughout the development life-cycle. In addition, we will apply a variety of human-centered engineering methods to reduce the overall development costs of INTACT. A collection of tools and methodologies are used in the design process. A number of these tools, but by no means an exhaustive list, are listed below.

- **Criteria Definition** - Measures of effectiveness, critical operational issues and other metrics
- **Heuristic Evaluations** - Review by HF experts to identify usability issues with the product (e.g., inconsistencies, insufficient system feedback, etc.).
- **Usability Evaluations** – Using one or more HF evaluation techniques including design walkthroughs, simulations and scenarios, questionnaires, experiments, performance evaluations to identify issues not only with usability, but usefulness of a product (i.e., does it provide all necessary functions in an effective manner).
- **Quick-Look Assessments** – Smaller-scale evaluation consisting of either a top-level review of a product (e.g., using heuristic techniques) or focusing on a known problem area (e.g., using usability evaluation techniques). When resources are limited, a quick-look assessment can be a cost-effective means of identifying areas needing improvement.

- **Design Recommendations** - Provide both GUI and feature oriented recommendations based on outcome of heuristic and usability evaluations in order to increase the usability and usefulness of the product.

7.3. Software Test and Evaluation

This task will involve system level testing of the software code. We will document our objectives and methods in a test plan, create testable requirements from the software specifications, and conduct and document the results of the testing. The emphasis is how well the tool meets the system specifications and the robustness of the code (e.g., bug-free). As part of this task we will perform a preliminary hands-on demonstration and collect subjective user feedback by administrating surveys on tool intuitiveness, usefulness, and ease of use.

As the coding process progresses, we will initiate a code freeze and commence the integration and test phase. It as at this point that we will test INTACT at the system level. This process is commonly referred to as verification and validation (V&V) and is intended to ascertain if the software meets the specified requirements. We will establish a testing protocol for a series of laboratory studies intended to measure the software's reliability and availability. This is a critical aspect of the software development process because it represents the first stage of measuring the operational utility of INTACT. At this stage of V&V, we aim to assess if the level of software development is high enough where its performability is adequate to allow operational functioning. Only after we complete V&V and confirm that the software is reliable will we initiate operator-in-the-loop testing.

7.4. Modifications to the DDD Testbed

The DDD is a unique distributed multi-person simulation and software tool for understanding command and control issues in a dynamic team environment. A full description is provided in Section 5.1.

The DDD provides an ideal testbed for quantitative assessment of the effectiveness of intelligent decision aids for command and control decisions. We can easily choose a configuration and a variety of scenarios from prior work that will create difficult, realistic command and control decision-making tasks. The challenge will be to identify the most promising DDD processes and decisions for intelligent coaching. Modifications to the DDD and the design of the DDD scenarios will be coordinated closely with the DCP and the operational needs statement to ensure that the DDD testbed is configured in a way that creates appropriate decision situations for testing coaching concepts. The DDD has been designed to support both individual and team command and control tasks. Although it is usually run with at least two decision makers, it is possible to configure it to create single-person tasks as we demonstrated in Phase I.

One of the most consistent findings during our data collection for the AWACS operational needs statement was the criticality of communications for the WD. Communication was viewed both as a key variable affecting operational performance as well as an indicator to an SD as to how well a WD is performing. As such, we view the addition of advanced communication capability for the DDD is a main objective for Phase II. This capability, in the form of voice recognition and generation will provide a wide range of development opportunities in to assess operational performance and to provide coaching assistance.

8. PHASE II EXPERIMENTAL PLAN

Our proposed Phase II experimental plan is comprised of two distinct research plans. The first is an attempt to validate the DCP, our theory-based model of coaching. The second will focus on assessing in utility of INTACT as an entire system. Both of these research plans are discussed in detail below.

8.1. Validation of Theory-Based Model of Coaching

In Phase II, we plan a series of five studies to validate the our Dynamic Coaching Protocol (DCP) concept. These studies represent the basic research aspects of our project. The STTR program is intended to serve as a vehicle for theoretical constructs to enter into commercial products. The studies described below represent the process by which we will validate theoretical hypotheses from university research prior to commercialization.

Study One will focus on the development of the multi-state state continua that will control the DCP model as shown in Figure 3. The study will represent accumulating, from the literature, the prototype task, display, and cognitive configurations that will represent the state values of each dimension. An empirical evaluation based on subjective and objective performance methods (workload measures, performance and so on) will be conducted to verify that the configurations do indeed produce the orderings on the continua that have been shown to exist in other studies. The algorithms will be tested for process validity by piloting with simulated data (factual and counterfactual simulations).

Study Two will focus on establishing the congruence principle as it relates to operational awareness, which is central to the DCP concept. While there has been work on the principle of congruence (see Hammond, 1996 for review), there has been little work aimed at combining the three dimensions (task, display, cognitive mode) in an effort to support cognitive awareness.

Study Three will focus on making changes to the continua that are suggested from the outcomes discovered in Study 2. Here, state values on each dimension will be evaluated for robustness in their effects on users (effect size assessment). In the case of small effect size outcomes, reconfigurations of state values will be made to produce reasonable discriminability among values.

Study Four will formally test the DCP model in a large-scale simulation venue. Results of the work will be submitted as evidence of the potential of our approach to assist decision-making.

Study Five will be rework the DCP model in light of the outcomes of the four previous studies. Here, modifications in the rule-base for invoking display interventions will be conducted. The objective of study five is to fine-tune the DCP concept and to replicate the predicted Coaching outcomes. The study five report will culminate with the publications of the software and documentation for the DCP intervention application.

8.2. INTACT Tool Validation

A fundamental question that we must answer is whether or not the software we develop for INTACT can reliably detect changes in the tactical environment and the operator dynamic state (see Figure 1). In addition, validation must address whether or not the intervention trigger can incorporate these perceived changes and communicate with the coaching module to provide proper recommendations. In this sense, validation of INTACT is a preliminary attempt to assess the tool's ability to coach (i.e., identify operational mistakes and provide accurate guidance).

8.2.1. Operator-in-the-loop testing

Operator-in-the-loop testing will be used to ascertain the operational utility of INTACT. This research will address the tools ability to improve mission performance, as well as its utility as a training tool. INTACT is intended to serve two purposes, enhancing performance and supporting training. Therefore, two distinct evaluations will be made—operational performance and training capabilities. Associated with both types of research, however, is a set of operational metrics, developed by Aptima, that will allow us to make accurate assessments of operational effectiveness on a number of levels. Following a discussion of these measurement issues, the operational performance and training capabilities evaluations are described in more detail

8.2.2. Measures for the INTACT Research Program

A major contribution to our selection of the DDD simulator as our primary testbed in Phase II was its associated research history and the development and validation of a set of automated measures that capture both individual and team performance, and team process. We plan to leverage ongoing efforts in our SBIR Phase II project, A System to Enhance Team Decision Making Performance, and employ a set of measures based on the development and use of mental models by the team members. We expect to deploy the full range of team measures, tailored for the AWACS WD task, during Phase II in experiments to evaluate the effectiveness of INTACT for enhancing operator effectiveness.

Numerous measures are available to assess team process and performance; however, very few measures to date have been automated. Table 6 shows the DDD measures that have been automated as a subset of team assessment measures. Aptima personnel, as a group, have pioneered automated collection of team measures using the DDD. Serfaty, Kleinman, and their colleagues (e.g., Kleinman and Serfaty, 1989; Kleinman, Pattipati, Luh, and Serfaty, 1992; and Saisi and Serfaty, 1988) have measured team behavior, cognition, and performance using a set of roughly 120 team measures.

Table 6. Team Assessment Measures

	OUTCOME PERFORMANCE	PROCESS/ INDIVIDUAL	PROCESS/TEAM	COGNITIVE/ WORKLOAD	TEAM STRUCTURES
AUTOMATED (DDD)	MISSION-SPECIFIC, SPATIO-TEMPORAL, ATTRITION	INDIVIDUAL DECISION-MAKING STRATEGIES	TEAM COORDINATION INDICES	INPUT TO SHARED MENTAL MODEL	TOPOLOGICAL/ GRAPH MEASURES, CONGRUENCE
OBSERVATIONAL (SUBJECTIVE)	MODIFIED EXISTING SCALES (ATPI)	MODIFIED EXISTING SCALES	TEAMWORK PROCESSES	SHARED MENTAL MODEL	
OBSERVATIONAL (OBJECTIVE)			COMMUNICATION, ANTICIPATION, & COORDINATION		DEPENDENCY INDICES
SELF-REPORT	SITUATION AWARENESS	SUBJECTIVE, ATTITUDINAL	SUBJECTIVE, ATTITUDINAL	TLX, SWAT	

The DDD measures, which have been validated in numerous empirical studies, fall into the following four categories plus a special *team structure* category.

- *Outcome Performance* measures assess dimensions of team performance. These measures in the DDD include the reward earned, team accuracy, and timeliness in processing information or items. We can examine components making up a team measure, such as individual team member performance.
- *Process/Individual* (taskwork) measures capture the mechanisms or processes by which the team attains its performance. DDD process measures include the team's degree of information seeking, resource utilization, and failure to perform certain tasks.
- *Process/Team* (teamwork) measures describe how the team strategy was accomplished. DDD measures include the total communication patterns among team members and conflicts caused by ineffective coordination. In addition to measures of coordination and communication presence and quality, we want to identify the specific aspects of the communication and coordination associated with error and error propagation. This requires collecting qualitative measures of team coordination behavior.
- *Cognitive and Workload measures* indicate the demands along various dimensions including time, mental effort, and psychological stress. Kleinman and Serfaty have pioneered the use of the Subjective Workload Assessment Technique (SWAT) (Reid, Singledecker, Nygren, and Eggemeir, 1981) and the NASA Task Load Index (TLX) to assess team workload and the dynamic redistribution of workload in teams.

8.2.3. Operational performance testing

We plan to conduct experiments to test the effectiveness of our coaching interventions for enhancing operational performance. We expect to be able to include more than one cycle of testing, so that we should be able to evaluate a variety of coaching interventions. The results of one experiment may influence the next. For example, if we find that a particular intervention worked only partially, and gain insight as to why, then we may want to include a modified and improved version of the intervention in the next experiment.

In each experiment we will conduct baseline trails followed by intervention trials. The baseline trials will provide us with benchmark data on performance on the DDD AWACS task, and how it is affected by scenario variables such as tempo and uncertainty. We can then repeat similar scenarios (varied to avoid anticipation) with INTACT operational and collect our experimental data. We will compare the benchmark data to the experimental data to assess the utility of INTACT. Of important note is the fact that we will conduct within-subjects experiments. This design affords us the ability to control for individual differences and will result in the most accurate assessment of the tool's utility.

Location of the experiments can be varied according to what is most convenient for the Air Force. It will be possible to install the DDD AWACS environment at almost any well-equipped computer lab. One possibility is to install it at Tyndall AFB with the support of the 325th Training Squadron. Another option is to leverage an existing relationship between Aptima and the US Air Force Academy. Under a previous program, we installed the DDD at the USAFA, in coordination with Captain Joey Hickox, a faculty member at the Academy. A third option would be to install it at or near Tinker AFB, allowing easy access to experienced AWACS subjects.

While it is important that the experiment subjects be qualified WDs and SDs, it is not essential that they be "intact" teams. We can use ad hoc teams in the experiment, as long as they have real world AWACS experience. The number of teams needed for an experiment will depend on the hypotheses being tested. Typically, 6-8 teams is a ball park number for DDD team experiments. It will be important not to ask the teams for an excessive time commitment for experiment participation, which may lead to a design that uses more teams for shorter periods, rather than fewer teams for longer periods.

8.2.4. Training capabilities testing

Experiments specifically intended to assess the training capabilities of INTACT will vary slightly from the operational performance testing. The primary difference is the need for longer duration experiments to assess the lasting effects of using INTACT. We will establish a repeated-measures, within-subject design to assess changes in performance over time as a result of using INTACT to support training. This experimentation requires a control group that follows the same testing schedule, however they will not be exposed to INTACT.

9. PHASE II TECHNICAL OBJECTIVES

In Phase I we defined a coaching model—Dynamic Coaching Protocol—that provides a framework to develop tools to support both operational training and performance enhancement. Furthermore, we demonstrated the feasibility of adding an **embedded coaching tool** to a simulated command and control environment. In Phase II we will validate our model and create a fully functional tool that will be designed to support operational performance in any tactical environment, including non-military applications. The Phase II INTACT will incorporate **enhanced student modeling techniques** to allow for **individually tailored pedagogy**. That is, the method of coaching will vary based on the individual users operational performance and the environmental (mission) conditions. Finally, we will, from day one, aggressively develop and pursue commercialization opportunities as defined in our commercialization plan (See Sections 10 and Appendix 3).

We have established five specific objectives to meet the Phase II challenge of developing a **software implementation of our intelligent coaching tool, INTACT**.

Phase II- Objective 1: Evaluate theory-based model of coaching and associated coaching interventions devised in Phase I.

In Phase I we created a theory-based model of coaching for command and control. Based on this model, we proposed and implemented coaching strategies to support operator performance in a simulated AWACS environment. In Phase II, we will conduct research studies, as specified by our Phase II experiment plan, to validate this model. While our model is theory-based and developed from the research literature, it is important that we confirm that our findings transfer to the command and control domain. Furthermore, we are interested in identifying individual differences that may moderate the overall effectiveness of coaching techniques.

Another aspect of this objective is to assess the robustness of the coaching models and methods. As stated in Objective 5 below, we are committed to transitioning INTACT to multiple domains. Therefore, it is imperative to ascertain if our pedagogical assumptions and usage are applicable and reliable in other situations. We plan to conduct studies in non-AWACS environments to assess the generalizability of our methods. This is intended to ensure that INTACT evolves as a multi-objective coaching tool and will help us develop a more complete commercialization plan.

Phase II-Objective 2: Expand Phase I operational needs statement and modify DDD testbed to address voice-based communication aspects of C2 team performance.

Command and control is an organizationally-based activity that requires individual decision makers to communicate and coordinate with other team members to ensure successful mission execution. In addition to this internal communication, many team members are required to communicate and/or coordinate with people and teams outside of their own organization. These communication and coordination responsibilities, both internal and external, were identified in our Phase I operational needs statement as a very difficult aspect of the AWACS operator's task.

We plan to follow-up our Phase I interviews with AWACS instructors and subject matter experts with a more detailed knowledge engineering process, focusing on communication and coordination responsibilities of the AWACS team. This information will be used to refine our coaching model as well as to identify candidate functionality to add to the DDD testbed. While voice communication issues were identified as important for AWACS C2 performance, they are not currently simulated on the DDD. Testing the operator's ability to respond to all voice requests in a timely manner requires adding both a speech synthesis and voice recognition component to the DDD. We plan to implement these capabilities and others that appear to offer the most value added to evaluate the utility of our intelligent coaching methods.

In Phase II, we intend to expand the AWACS operational needs statement, as well as move beyond the AWACS domain. An overall goal of this effort is to generalize our coaching methods and we will select and study a second command and control organization to evaluate and apply our methods. Our plan is to identify and study a non-Department of Defense organization such as an emergency dispatch station. We plan to create a command and control team operational needs statement for this domain and demonstrate how INTACT can be used to support operational performance and the training process. This will allow us to **identify viable domains for transition opportunities**.

We also plan to explore the utility and need for adding a query function to the coach that would allow the user to enter into a dialogue with the coach to solve a problem. In Phase I we focused primarily on how an embedded coaching tool could provide automatic operational support to the user. That is, the coach was intended to run in the background and intervene as necessary. The additional functionality proposed here would allow the user to be proactive in seeking assistance with tasks that he or she anticipates or finds to be challenging.

Phase II-Objective 3: Build a fully functional version of our embedded coaching tool–INTACT –to support operational performance and training.

We plan to adapt and enhance our current software to yield a **self-standing** Intelligent Tactical Coaching Tool (INTACT). INTACT will be a unique tool that builds upon basic research in coaching strategies to address specific operational needs of the command and control operator. Tool building is the major focus of our Phase II effort and is comprised of several subtasks:

A) **Specification Development** – Based on our experience in Phase I and the results of our continuous knowledge engineering process, we will author a document detailing tool and end-user requirements and system specifications. We will consider requirements for tool integration and interoperability with the DDD and other simulators in Phase III. More specifically, we will investigate and incorporate, as feasible, High Level Architecture (HLA) for INTACT to support interoperability. One possibility will be to integrate INTACT with the C3STARS high fidelity AWACS simulators at Brooks, AFB.

B) **Software Development** – All major features of the proposed design, including those implemented in Phase I and those defined in subtask A, will be implemented. The major features that the software will support are: *real time* coaching in a C2 environment; *adaptable coaching techniques* to match operators' decision-making and battle-management needs; and *collection of performance measures* that allow INTACT to provide performance and comparative analyses to the operator at mission completion.

C) **Test and Evaluation** – This task involves system level testing of the software code. We will document our objectives and methods in a test plan and document the results of the testing. The emphasis is on how well the tool meets the system specification and the robustness of the code (e.g., bug-free). As part of this task we will perform a preliminary hands-on demonstration and collect subjective user feedback on tool intuitiveness and ease of use.

Phase II-Objective 4: Demonstrate the operational effectiveness of INTACT.

We will **demonstrate the utility of INTACT** by conducting operator-in-the-loop experimentation. We plan to test intelligent coaching concepts using the Distributed Dynamic Decision-making (DDD) simulation environment. The DDD can be configured to represent command and control tasks for dynamic scenarios in a variety of environments and has previously been used to study decision making for Weapon Directors on board AWACS aircraft (MacMillan et al., 1997). The DDD will provide an ideal testbed for quantitative assessment of the effectiveness of intelligent coaching for command and control decisions. We will create difficult, realistic command and control decision-making tasks and assess operator and mission performance with and without the assistance of coaching. Repeated measure studies will be conducted to assess the training benefits of INTACT.

Phase II-Objective 5: Commercialize INTACT.

While the central goal of Phase II is to create an INTACT tool and apply it to a command and control domain, a portion of our effort will be aimed at ensuring that INTACT has widespread applications for both military and commercial users. To enable this application diversity, we will solicit input from potential post application users early in the needs analysis, design, and validation process. In addition we will develop a demonstration methodology that can be used to cultivate a commercialization success. We will leverage both Aptima's and the University of Georgia's success in marketing the benefits of decision support systems in real world systems.

The proposed commercialization plan will commence following the project kickoff meeting and continue throughout the entire Phase II period. Within the government, we have identified four specific applications for INTACT. Our primary Air Force market is the AWACS training and operational communities. Other government opportunities include Navy Aegis CIC operations, air traffic control, and the U.S. Coast Guard. Potential commercial venues include the lucrative educational software industry, developing corporate decisionmaking support tools, and embedded coaching for driver education.

9.1. Anticipated Benefits

Our goal is to develop a theory-based, model-driven, real-time, intelligent coaching tool for command and control operations. INTACT is being designed to address the current operational challenges associated with maintaining high-quality performance in complex multi-task environments and the shortage of qualified people to perform these difficult tasks. We envision providing the operator with an embedded tool that will reduce the workload burden associated with these complex environments and provide customized pedagogy to improve training processes.

The tool is motivated by the need to provide both real-time mission and instructional training support to command and control operators. The need for such a tool is likely to increase in coming years as technological advances will increase the scope of responsibilities (span of control) each command and control team must address at the same time as staffing reductions limit the number of operators available to perform those responsibilities. Rather than develop two separate tools, our approach is to develop a single tool that is able to support the tactical decision maker in both operational and training situations. The most obvious benefit of this approach is its direct correspondence to the military maxim of "Train like we fight."

The coach is designed to improve performance in the short-term as well as aiding learning in the long-term. The development process we have proposed will enable us to answer specific questions about the best methods of providing coaching to operators and evaluate the role of individual differences in the effectiveness of these methods. We believe that the methods we develop will not be limited to the command and control domain, AWACS, that we selected for demonstration in Phase I. In Phase II, we will select and study a second domain and demonstrate how our coaching methods could be used to support performance and training in that domain.

10. COMMERCIALIZATION PLAN

As shown in our Phase II Technical Objectives (see Section 9), our effort will focus on developing the INTACT software based on the proof-of-concept prototype developed in Phase I, our theory-based model of coaching (DCP), and our Phase II software functional requirements analysis work. By the completion of Phase II, INTACT will be an embedded tool intended to reduce the workload burden associated with complex, information-rich, multi-task operational environments.

While we will demonstrate the utility of the tool within the AWACS environment, our development effort is intended to allow these methods to be applied to a variety of applications. INTACT is also viewed as a tool capable of being used to aid in the training of new warfighters by providing an embedded coach for simulation based training. The coach is intended to ensure that a student is able to complete simulated mission tasks correctly, even when unsupervised during extracurricular practice sessions. Again, we will focus on AWACS in Phase II, but generalizability of our tool is of paramount importance.

One of the practical benefits of viewing and designing this tool as both an operational and training aid is that it is in line with the goal of "fight like we train," in that the same tool will be used in both settings. This approach also will impact our ability to leverage these efforts in future research and development. Specifically, we will be able to apply INTACT and the theory-based model of coaching to a wide range of decision support and training programs, both of which are important topics of study in both government and commercial settings. Below, we have enumerated several post-Phase II applications for INTACT and the associated coaching methods. In addition, we have provided a copy of our formal commercialization plan, submitted as part of our Phase II proposal, as Appendix 3.

10.1. Potential Post Applications

The implementation of software agents in intelligent coaching devices is a technology that lends itself naturally to commercialization opportunities. Any environment that depends on highly reliable human decisions involving dynamic resource allocation under conditions of uncertainty is a candidate for intelligent coaching support.

Within the government, we have identified four specific applications for INTACT. Our primary market is the AWACS training and operational communities. We have laid out a programmatic development approach that transitions INTACT from research tool to a real-world embedded performance support tool. This plan is presented in Table 7. As can be seen, we will work with the Air Force's research organizations to validate our tool and introduce it to the operational community through its training organizations. This approach will build credibility and buy-in from the operational community and will ease transition on the AWACS aircraft itself.

Table 7. Programmatic Development Approach for AWACS-Based INTACT

Application	Justification	Outcome
C3STARS, Brooks AFB	Center of excellence for AWACS C2 research	Conduct human-in-the-loop experimentation to validate and refine coaching tool
Tyndall AFB	Initial training center for weapons directors, air battle managers, and ground radar controllers	Demonstrate value of embedded coach for initial training as a supplement to academic training programs
Tinker AFB	Home of the AWACS Wing and advanced simulator-based training programs	Illustrate coach ability to support advanced mission and tactical training, as well as provide real-time mission support
Situation Display Console on E-3 Aircraft	Ultimate outcome of development project is to create tool to support the warfighter during missions	Prove mission utility to real-time decision support and significance of a "train like we fight" approach to AWACS operator development

The remaining three government applications are described below and are followed by descriptions of potential non-government INTACT implementations.

10.1.1. Government Opportunities

Navy ATD for Advanced Embedded Training Systems: This established Navy program is intended to design training and decision support tools for Aegis Battle Staff. This program also presents an opportunity to transition into commercial application by working with Lockheed Martin's Advanced Technology Laboratory. Their mission is similar to the ATD in that they are building tools to support the next generation of battle staff for the DD-21—the destroyer of the 21st century—currently in the concept development phase. A primary goal of DD-21 is to conduct combat operations with a smaller battle staff, without requiring excessive training. INTACT and the INTACT design process can aid in the development of these support and training systems.

Air Traffic Control (ATC): INTACT can be used to support current ATC operations by advancing the training process and providing real-time decision support. A more critical FAA need may be dealing with the free fight environment. Most would agree that free flight will fundamentally change the way ATC is managed, requiring new methods of training and increasing the need for embedded performance support. Potential sponsors include the Volpe Research Center, the FAA Technical Center, and the Air Traffic Surveillance Group at Lincoln Laboratory.

U.S. Coast Guard: INTACT can play dual roles for the Coast Guard. Internally, Coast Guard search and rescue and drug interdiction operations can be supported by embedded performance support, specifically the command and control aspects of these tasks. Externally, we can work with the Coast Guard to develop training tools for commercial maritime operations focusing on understanding and following the "rules of the road." A potential sponsor for these efforts is the USCG Research and Development Center in Groton, Connecticut.

10.1.2. Commercial Opportunities

Educational Software: The implementation of technology as an instructional aid has become quite common in the classroom setting. Specifically, intelligent tutor programs have become a permanent fixture in many classrooms across the country. Such programs have been lauded for their ability to facilitate the teaching process in a number of ways. With the help of intelligent tutors, lessons can be tailored to students' learning styles (e.g., lecture, interactive, video examples), and students can therefore be free to learn at their own pace. Furthermore, students may benefit from multiple sources of knowledge in addition to their instructors and textbooks, as intelligent tutoring programs may incorporate knowledge from a number of experts in any given area. We believe that the Dynamic Coaching Protocol (DCP) and the associated coaching mechanisms can aid the enhancement of these existing systems.

Corporate Decision Making: An emerging area for commercialization may be tools that can support decision making in corporate settings. For example, tools to support the proper allocation of resources during project management or computer-based simulations that can help train mangers make better strategic and tactical business decisions would appear to have value within the corporate environment. If one views these examples as not fundamentally different than those faced by a battle manager executing command and control, then the application of INTACT to this venue is readily apparent.

Driver Education: Our Phase I demonstration of INTACT was highlighted by the fact that we were able to embed a coaching tool within an existing simulation device. In Phase II we will improve the interaction between INTACT and resident simulators through the exploration of HLA. This ability to add imbedded coaching would be useful as a means to support driver education programs.

11. REFERENCES

Elliott, L.R., Dalrymple, M.A. & Neville, K. (1997). Assessing performance of AWACS command and control teams. *Proceedings of the Human Factors and Ergonomics Society 41st Annual Meeting*, pp. 172-176.

Goldstein, I.L. (1993). *Training in Organizations*. Pacific Grove, CA: Brooks/Cole Publishing Company.

Hammond, K. R. (1996). *Human judgment and social policy: Irreducible uncertainty, inevitable error, unavoidable injustice*. New York: Oxford University Press.

Hoffman, R. R., Crandall, B., & Shadbolt, N. (1998) Use of the critical decision method to elicit expert knowledge: a case study in the methodology of cognitive task analysis. *Human Factors 40(2)*, 254-276

Kaempf, G. L., Klein, G. A., Thordsen, M. L., & Wolf, S. (1996). Decision making in complex command-and-control environments. Human Factors and Ergonomics Society, 38(2), 220-231.

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

Kleinman, D., Pattipati, K. Luh, P., and Serfaty, D. (1992). Mathematical models of team performance: A distributed decision-making approach. In R. Swezey and E. Salas, (Eds.) *Teams: Their Training and Performance* (177-218). Norwood, NJ: Ablex Publishing Corporation

Kleinman, D.L. & Serfaty, D. (1989). Team Performance Assessment in Distributed Decision-Making, in Gibson et al. (Eds.), *Proceedings Symposium on Interactive Networked Simulation for Training*, Orlando, FL.

MacMillan, J., Serfaty, D., Young, P., Klinger, D., Thordsen, T., Cohen, D. & Freeman, J. (1997). *A System to Enhance Team Decision-Making Performance: Phase I Final Report*. Woburn, MA: Aptima, Inc. AP-R-1102.

Pete, A., Pattipati, K.R., and Rossano, C. (1991). Distributed Binary Detection with Different Local Hypotheses. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics* v 3 Oct 13-16 1991, p 2023-2028.

Reid G. B., Singledecker C. A., Nygren T. E., and Eggemeir, F. T. (1981) Development of Multidimensional Subjective Measures of Workload. Proceedings of the 1981 IEEE International Conference on Cybernetics and Society, pp. 403-406, Atlanta, GA.

Saisi, D. L., and Serfaty, D. (1988) Model-Based Human Interface Design for Distributed Tactical Command and Control, *JDL Command and Control Research Symposium*, Monterey, CA.

Serfaty, D. (1996). Adaptive architectures for command and control: An overview. In *Proceedings of the 1996 Command and Control Research and Technology Symposium*, Naval Postgraduate School, Monterey, CA.

Serfaty, D., and Entin, E.E. (1995) Shared Mental Models and Adaptive Team Coordination, *Proceedings of the International Symposium on Command and Control research and Technology*, pp. 289-294, June 1995, NDU, Washington, DC.

Serfaty, D., Entin, E.E., and Johnston, J. (1998). Team Adaptation and Coordination Training. To appear in *Decision Making Under Stress: Implications for Training and Simulation,* Eds. J. A. Cannon-Bowers and E. Salas, Washington D.C.: APA Press.

Serfaty, D., and Kleinman, D.L. (1985). Distributing Information and Decision in a Team. *IEEE 1985 Proceedings of the International Conference on Cybernetics and Society. 1985*, p 767-771.

Simon, H. (1987) Two heads are better than one: The collaboration between AI and OR. Interfaces, 17 (4), 8-15.

Wang, W.P., Luh, P.B. Serfaty, D., and Kleinman, D.L. (1991) Hierarchical Team Coordination in Dynamic Decisionmaking. *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics* v 3 Oct 13-16 1991, p 2041-2047.

Wickens, C. D., & Carswell, C. M. (1995). The Proximity compatibility principle: Its psychological foundation and relevance to display design. *Human Factors, 37*(3), 473-494

APPENDIX I:
DYNAMIC COACHING PROTOCOL

Advanced Human Resource Project (AHRP) Laboratory,
University of Georgia

OVERVIEW

This proposal outlines the theoretical approach behind an intellectual aide that serves to bridge instructional technology and decision support approaches in creating intelligent coaching systems. The approach is grounded in a technology that can monitor task and mediation properties in order to help provide levels of operational awareness that match operational demands. The objective is to build a framework called Dynamic Coaching Protocol (DCP), which will lead to the development of adaptable software agents that can help guide expertise in military operations. An explicit metatheory will provide coaching principles that can be applied to a variety of operational contexts, whereas the triggering components of the coaching system will be scaled to the operational contexts at hand. The goal is to create a configurable coaching system that leverages context independent theoretical principles needed for a federated coaching software system.

A strong adaptable framework has been chosen in which to develop the cognitive engineering perspective illustrated in this proposal. First, a nomological network of ideas that forms the basis of an adaptable metatheory on human decision making and judgment is outlined. This network includes an overview of the metatheory of Probabilistic Functionalism, and its importance in specifying design options for optimized intellectual support is defined. Secondly, a corresponding adaptable methodology for implementing support through the use of an intelligent agent is outlined.

Background on Intellectual Support

The implementation of technology as an instructional aid has traditionally been restricted to the classroom setting. Specifically, intelligent tutor programs have become a permanent fixture in many classrooms across the country. Such programs have been lauded for their ability to facilitate the teaching process in a number of ways. With the help of intelligent tutors, lessons can be tailored to students' learning styles (e.g., lecture, interactive, video examples) (Crynes & Hawley, 1995; Srisethanil & Baker, 1995), and students can therefore be free to learn at their own pace. Furthermore, students may benefit from multiple sources of knowledge in addition to their instructors and textbooks (Rush & Wallace, 1997; Schofield, Eurich-Fulcer, & Britt, 1994), as intelligent tutoring programs may incorporate knowledge from a number of experts in any given area. Finally, due to the amount of time students spend interacting with the tutor programs, instructors are free to spend more time assisting students at the individual level.

The intelligent tutor programs presently in use are based on a traditional academic model. These programs are designed for novice users with little or no experience in the topic area of interest. Therefore, the purpose of these programs is to develop a knowledge base in an area in which no knowledge currently exists. While technology of this kind has found its place in the classroom, technology of a different nature is necessary in more applied settings in which users have already developed a foundation of knowledge. Specifically, a different form of intelligent technology is necessary for use as a training aid to assist experienced individuals in enhancing their performance. For example, training aids may be utilized to help users overcome limitations or to fine-tune their skills in a specialty area.

A comprehensive review of the literature in the domain of intelligent technology is necessary in order to demonstrate the need for the development of training aids tailored for use in applied settings. While the domain of intelligent technology is quite diverse, several common themes appear to characterize the majority of scientific research. The present review will highlight some of these areas of focus and provide descriptions of prototypes developed for the use in applied settings.

Decision Aids and Cognitive Effort

Recent research has investigated the implications that the use of decision aids may have for cognitive processing. For example, the design of the decision aid may impact the choice strategy employed by the user. Todd and Benbasat (1994a, 1994b) posit that a method of processing may be invoked by altering the effort requirements of decision aids. Specifically, if the support provided by a decision aid makes the use of a more difficult, more accurate choice strategy as easy as the use of a simpler, less accurate strategy, then the use of the decision aid is likely to enhance decision quality. Otherwise, decision quality may be compromised, as users tend to conserve cognitive effort by opting for the simpler, less accurate strategy (Benbasat & Todd, 1996).

The utilization of decision aids may have a negative influence on the manner in which decision makers approach aided tasks. Although the technology is designed to facilitate the process by which users arrive at their decisions, research suggests that inexperienced decision makers may rely excessively on the decision aids (Glover, Prawitt, & Spilker, 1997; Todd & Benbasat, 1994a, 1994b). This over-reliance may influence the decision makers to apply a mechanistic approach to the aided tasks rather than becoming actively involved in the decision making process. Ultimately, this lack of cognitive participation may inhibit the user from actually learning the task at hand. Furthermore, it must be noted that the user may not realize that he or she has not adequately learned the task. This misunderstanding may have deleterious consequences once the user is placed in actual situations in which he or she is expected to have the knowledge necessary to complete the task. This problem is likely to be severe due to the types of tasks, which incorporate such technological training (e.g., air traffic control).

One way to avoid the negative consequences of a passive approach to learning is to implement decision aids, which require active participation. Such programs may be superior to typical decision aids in that they require the user to "combine and process information in order to produce a meaningful decision or solution" (Glover et al., 1997, p.238). The necessity of cognitive involvement may therefore be a critical component of successful decision aids.

Simulation and the Development of Mental Models

Through the use of decision aids, intelligent tutors, simulations, and other technological formats, users may be introduced to different types of situations that are likely to be encountered when actually performing a given task in real world conditions. Specifically, these forms of training have the advantage of presenting users with potentially dangerous predicaments requiring quick decisions (e.g., loss of engine power on an airplane) without the risk of severe consequences for making mistakes (Gonzales & Ingraham, 1994). Users can learn to determine what information is necessary for their decision, how to execute the decision once made, and ultimately realize the results of their actions.

The previously described cycle of events may repeat itself through a number of iterations before the user has formed a knowledge base strong enough to adequately function under actual, rather than simulated conditions. This knowledge base may be better understood as a mental model (Haarbauer, et. al., 1999). Mental models are representative of an individuals' expectations for a given situation. Therefore, through the application of mental models, individuals can filter out extraneous details and more easily focus attention towards the critical information necessary to solve a specific problem (e.g., Kontogiannis, 1996; Silverman, 1995). As a result, the speed, effectiveness, and quality of decision making may be enhanced. It must be noted that for novice users, the utilization of simulations and/or intelligent technology aids in model formulation, as no knowledge base exists for the situation of interest. In contrast, for more experienced users the interaction with such programs aids in model selection or specification (Benbasat & Todd, 1996). More specifically, these tools allow experienced users to fine-tune and adjust their mental models for variations of a given situation as well as determine which of these distinct models is appropriate when such variations are encountered.

Despite the benefits of the formation and implementation of metal models, it is important to take the context of a situation into consideration. As Turner (1998) explains, for humans and animals, behavior is context dependent; therefore, intelligent agents should also display context-sensitive behavior. He introduces the concept of context-mediated behavior, or the notion that an intelligent agent should have and be able to incorporate explicit contextual knowledge (i.e., contextual schemas). Turner (1998) further explains that "each contextual schema contains both descriptive knowledge about a particular context and prescriptive knowledge about how the agent should behave in that context " (p.308).

While intelligent technology such as decision aids or intelligent tutors are learning tools rather than autonomous agents, the concept of contextual schemas is still applicable. Specifically, a form of intelligent technology could have such descriptive and prescriptive knowledge, however, it could use this knowledge as a guide through the instructional or training process. As a result, users could learn to identify the context of a given problem or situation, an then incorporate information from contextual schemas to formulate predictions about unknown or ambiguous details (Turner, 1998), ultimately aiding in the speed and quality of the decision-making process (Ozturk & Aamodt, 1998). Essentially, both the user and the technology would utilize mental models through their bi-directional interactions.

This notion of context-based knowledge on the part of intelligent technology or decision aids has been recently examined. Based on the idea that solution quality and efficiency can be enhanced through the incorporation of context into such technology, Ozturk and Aamodt (1998) introduced a context model for case-based reasoning in decision-support systems. These researchers posit that hypothesis generation is a critical part of the diagnostic processes of any form of decision aid. In any situation, hypothesis generation involves abductive reasoning, or taking knowledge of past problems or similar situations into consideration. This dependency on abductive processes necessitates the existence of a mental representation or a "familiar prototype" for use as a starting point for hypothesis generation (Hatano & Inagaki, 1992). Taken together, the use of context-based simulation technology appears to be a well-suited tool for aiding in the development and refinement of mental models and ultimately enhancing the speed and quality of decision making in critical situations.

Requirements for Successful Intelligent Technology

After careful review of the literature, it is possible to identify some characteristics necessary for the effectiveness of any intelligent technological system. First, an intelligent system must have the ability to take users' prior performance into consideration and to tailor "lessons' to the knowledge level and cognitive styles of the user (e.g., Fleming & Horwitz, 1996; Srisethamil & Baker, 1995). Specifically, this adaptability is the critical aspect and defining component of intelligent technological systems. Moreover, adaptability is necessary for true bi-directional interaction between the user and the system to take place, for without this capability, information exchange would be unidirectional. In tandem with the notion of adaptability is abductive reasoning. As previously described, abductive reasoning is the ability to take previous knowledge and past experience into consideration during the reasoning process. Therefore, in addition to taking the specific characteristics and history of the user into consideration, the system must also incorporate context-based knowledge into the guidance of the user.

In addition, the organization and presentation of the knowledge to the user must be in a manner conducive to learning (e.g., Vasandani & Govindaraj, 1995). Otherwise, the tool may actually hinder rather than facilitate the learning process. Similarly, an intelligent tutor system must generate a coherent representation of the context. This is necessary for the accurate development and subsequent application of mental models.

In summary, Chu, Mitchell, and Jones (1995) provide a list of components necessary for an intelligent system to be useful in complex dynamic environments. The first component is a domain expert module, which includes the knowledge to be taught to the user as well as a standard against which the user may be compared and evaluated. Second, an intelligent system should contain a student model or a record of the user's performance. The student model is dynamic in nature as it includes the tutor's evolving assessment of the user's state of knowledge. The third component is the pedagogy module, which is responsible for the selection and presentation of instructional material. Fourth, intelligent tutoring systems must have an interface component, which represents the interactions between the tutor and the user. A control component is necessary to coordinate the other previously described components. The final component is a simulated environment which is necessary for real-time supervisory control as well as a dynamic context in which the user can practice newly learned skills.

Prototypes and Applications

Because each intelligent system is unique, the best way to gain a clear understanding of their components, parameters, applications and areas for improvement is to take a closer look at several prototypes. The following sections include descriptions of several systems, highlighting details, which exemplify the previously discussed areas of interest (i.e., cognitive issues, simulations, mental models, and context).

The Contingent Operator Stress Model (COSMO).

The Contingent Operator Stress Model (COSMO) is a decision-making model which integrates strategies of recognition and analysis in coping with stressful emergencies (Kontogiannis, 1996). Specifically, COSMO was designed to aid in the identification of multiple decision skills and the cognitive activities underlying them to form a foundation for the design of emergency response scenarios as well as the generation of training strategies. The COSMO framework supports three types of decisions including a) diagnosis and assessment, b) choices among alternative options, and c) the scheduling of tasks and monitoring of progress. The user must proceed through seven decision stages, each associated with different cognitive processes. These stages include 1) making an early appraisal, 2) formulating the problem, 3) recognizing and selecting an option, 4) re-assessing the situation, 5) evaluating options/goals, 6) planning the task, and 7) implementing and monitoring the task.

The COSMO system has been demonstrated in the context of the nuclear power emergency. For example, a situation may be presented in which there is a small break in a reactor coolant of a pressurized water reactor plant. The operator is faced with a sequence of cues on simulated control panels, and he or she must gradually narrow down alternative explanations for the problem at hand. Specifically, the incorporation of the COSMO framework aids operators in understanding how to use available information to "both distinguish possible faults at different stages of the emergency situation and recover from mis-diagnosis made at earlier stages " (Kontogiannis, 1996, p. 89). Furthermore, operators can learn to identify and subsequently prevent some common stress-induces sources of bias, such as cognitive tunnel vision (i.e., the failure to reconsider initial assessments of the situation) or groupthink (i.e., the tendency to concede to the opinions of the leader without taking alternative views or ideas into consideration; Janis, 1972). In addition, the COSMO system can be modified to incorporate different emergency simulations by varying the type and amount of information uncertainties (e.g., Mumaw & Roth, 1992).

As a training tool, this system is able to address the cognitive issues that are inevitable in stressful situations through the breakdown of the seven steps outlined earlier. Furthermore, COSMO facilitates the development and refinement of mental models by including modifications for different details of a given emergency situation. Finally, the COSMO system has the advantage of dealing with team-based situations, as evidenced by its attempts to keep communication lines open thereby reducing the likelihood of groupthink.

The Automated Adaptive Training System (AATS).

The automated adaptive training system (AATS) is an example of an Intelligent Instructor Support System (IISS). Unlike typical intelligent tutoring systems, IISS provides tools for "automating various facets of an instructor's task in existing simulators through the application of artificial intelligence" (Gonzales & Ingraham, 1994; p. 863). Specifically, IISS serves to provide feedback to the user as well as determine the sequence of training exercises to be presented.

One example of an AATS has been applied to the Army M1 main battle tank gunnery training. Upon completion of a tank gunnery training exercise, the student's scores are received by the AATS. The system then determines if the student has successfully passed (or failed) the task and subsequently updates the student model. Similar to the student model described by Chu, Mitchell, and Jones (1995), the student model represents the users current state of knowledge. AATS then chooses an advanced or remedial exercise to be presented to the student.

AATS is made up of a number of components, including an expert knowledge base, system interface, student evaluator, instructional model manager, and a student model manager. The expert knowledge base is incorporated into the instructional model manager, the most important component of the AATS. This module is responsible for determining the progression and/or remediation of exercises, and it contains the user's current goals and active plans. There are a number of exercise parameters considered by the AATS, such as target conditions, target type, visibility conditions, and scenario distractions. The values of these parameters can be varied to create exercises tailored to a specific user. For example, if a user fails an exercise that contains two types of targets (e.g., main gun targets and machine gun targets), the next remedial exercise may contain only the type of target with which the user is having the most difficulty.

This system clearly exemplifies the incorporation of a student's past performance into an intelligent system. The sequence of exercises is dynamically determined by the system through the evaluation of performance on each specific exercise. In other words, the student model is continuously updated as it is used to determine the exercise appropriate for a user at a particular time.

Georgia Tech Visual and Inspectable Tutor and Assistant (GT-VITA).

The Georgia Tech visual and inspectable tutor (GT- VITA) provides a "protected and guided environment in which a student can practice operational skills, including those applied to rare and catastrophic system conditions " (Chu, et al., 1995; p. 1055). As previously discussed, Chu and colleagues (1995) presented a list of components required for intelligent tutoring systems, including a domain expert, student model, pedagogy, user interface, control component, and a simulated environment. The authors present GT-VITA as a system, which meets these requirements.

The GT-VITA system has been implemented by NASA satellite ground control. The lessons presented by GT-VITA are in a realistic context through the use of simulation devices. Each lesson is composed of an instructional strategy, instructional content, and a scenario context. The student proceeds through a sequence of lessons beginning of increasing complexity, with remedial lessons incorporated when necessary. Specifically, in early lessons, the student is given information about NASA system objects (i.e., declarative phase), followed by lessons including demonstrations of particular procedures (i.e., procedural phase). Later lessons require the student to perform these procedures as the system progressively reduces the amount of assistance provided (i.e., operational phase). Upon completion of training, the student should be able to successfully support real-time command and control.

NASA conducted an empirical evaluation of the GT-VITA system utilizing sixteen participants representing a cross section of NASA Goddard Space Flight Center ground control personnel. Following the first phase of training (i.e., the procedural phase), all participants were able to demonstrate sufficient knowledge of the structure and components of the NASA network. After completion of the procedural phase of training, participants were able to articulate the procedural steps required for specific tasks. There were two tasks with which the participants had some difficulty describing, however satellite ground controllers are not required to memorize the procedures for these tasks in actual operations. Mixed results were found for the assessment of the operational phase. Specifically, it appears as if participants were still in the process of learning to apply their knowledge in the context of real-time operations at the time of assessment. However, following subsequent examination, NASA concluded that the GT-VITA system provided effective training for novice satellite controllers. Furthermore, participants reported positive reactions as well as the desire to continue training with the system.

Taken together, the GT-VITA system incorporates many of the desirable aspects of an intelligent system. Most importantly, the system aids in the gradual development of mental models, which can be later used to facilitate decision making. Furthermore, the emphasis on context and real-time simulation allows the user to calibrate his or her mental models for use in novel situations.

Real-Time Coaching

As can be seen by the multitude of different applications, the domain of intelligent technology is vast. Systems have been developed for use in any number of capacities. Although these tools have demonstrated their worth for the purposes of training and instruction, they are still held captive within the bounds of the traditional academic model. As previously discussed, most forms of intelligent technology are designed for the novice user who lacks a foundation of knowledge in the area of interest. Through the use of intelligent technology, the user hopes to develop a mental model of a specific task domain. In an ideal situation, the user would also be able to fine-tune his or her mental models and overcome limitations. Although the prototypes described above are designed to accomplish this task, it must be noted that the potential for refinement is limited due to the initial amount of knowledge brought to the training by the novice user. Therefore, it is proposed that intelligent technology of a different kind is needed for use with educated and experienced users with the desire to build upon their existing knowledge and skills to reach a higher level of specialization and expertise.

These goals may be attained through the implementation of real-time coaching. Specifically, real-time coaching can be understood as a combination of intelligent simulation and training technology with apprenticeship (Lesgold, Eggan, Katz, & Rao, 1992). Like other forms of intelligent simulation, users are performing "real" tasks rather than exercises. Namely, the training provides a simulation of the environment in which complex problem solving and non-trivial practice can occur. This active participation in the learning process is thought to maintain motivational levels and facilitate deeper cognitive processing. Training simulations incorporate contextual information, therefore yielding the benefits previously discussed (e.g., providing both descriptive and prescriptive information (Turner, 1998), development of mental representations or a "familiar prototype" (Hatano & Inagaki, 1992)). Furthermore, coaches can provide users with knowledge in the context in which this knowledge would later be applied.

Real-time coaching systems must also have the ability to monitor an operator's progress in detail. The automated nature of this process may be superior to the monitoring abilities of an actual human coach. Paired with "expert" level information, the incorporation of this information provides the system with "all of the knowledge needed to aide reflective opportunities for a user of the system" (Lesgold et al., 1992, p. 50).

However, despite the benefits and distinguishing features of coaching from other forms of support, little work has been done regarding this hybrid intelligent technology. One critical aspect in distinguishing coaching from other support techniques is that coaching is conceived as an operational awareness tool as opposed to a pedagogical device or a mechanism that aides operators with the computational demands of a task. Thus, coaching is best viewed as an agent that monitors the operational awareness manifested by a user for given environmental conditions.

Describing Operational Awareness

A conceptual framework for how an expert military operator processes tactical information in executing judgment can be expressed as a state analysis problem. A process model of judgment provides a framework for examining the details associated with the rules and mechanisms that are the antecedents to judgment. The process model is a representation of the judgment policy itself, where elemental features of the policy are exposed.

One conception of the state system is based upon that adopted in the theory of dynamic systems and intelligent automata (Aribib, 1972). The state concept is used here to discuss the judgment process as a sub-set of biological systems (Bunge, 1980). The emphasis is on knowledge states or cognitive awareness states.

A complex situational awareness system may be viewed as existing at any given point in time in one large number of states. From a theoretical viewpoint the number of states can be infinite. The state space is defined as a set of all states the system can be in, an is represented by an n-dimensional array made up of functional ranges for each property of the system. A particular state is defined as a point in this space, which is represented by a pattern of values that correspond loosely to what is sometimes called the "estimate of a situation". In this case, the properties that define situational awareness can be considered system vectors. From a practical standpoint in modeling situational awareness, the number of properties or indicator variables, is kept relatively low. Only variables thought to be important in determining the system's behavior are considered.

An additional feature that is important to consider is that in any dynamic system the state space will be in flux. It is unlikely to be either valuable or possible to consider transient states that endure only briefly. Further, one may argue that as expertise develops, the ability to quickly categorize information becomes better. One might expect that this would lead to system stability and reductions in the fluctuation of the system.

Figure 1 presents a finite state model of tactical judgment. There are essentially four basic components shown in the model: (1) tactical environment, 2) the human operator, 3) the state of situational awareness, and 4) the action space. The integral idea is one of recognizing and processing meaningful patterns of data in the tactical environment, and mapping the patterns of meaningful data over to the action space for the appropriate judgment or decision.

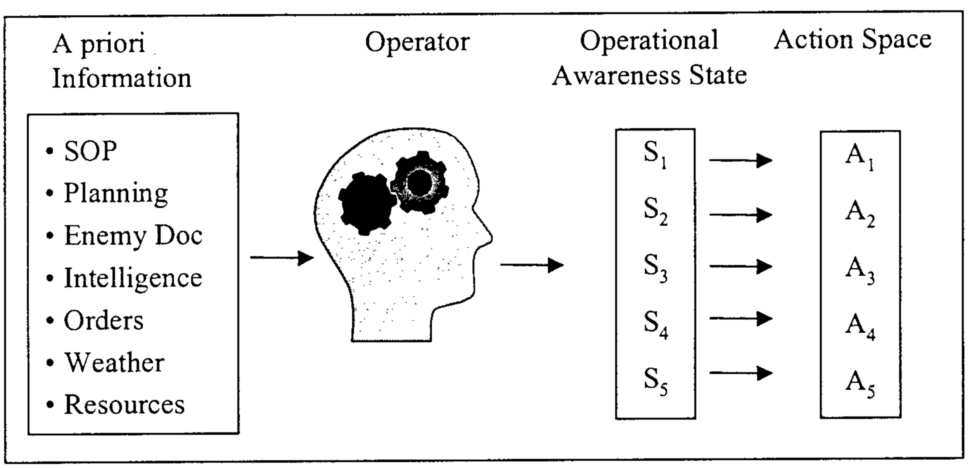


Figure 1. State judgment model showing four components: Tactical Environment, Operator/Performer, Awareness Vector, and Action Space.

In the state process model, the tactical environment can be partitioned into two sub-components. The first sub-component may be envisaged to contain a priori information that remains relatively static during the battle. This data category can describe an almost unlimited amount of information as long as it is historical in nature. For example, it can include past operational intelligence, historical knowledge of enemy military doctrine, formulated battleplans, orders, fragmentary orders, and standard operating procedures. This a priori knowledge is the supporting context for real-time decision making.

The second sub-component of the tactical environment symbolizes real-time data elements that are evolving around the observer. These real-time activities occurring on the battlefield and in communications, in theory, condition or modify dynamically what is known about the current situation. For example, the validity of past intelligence information is strengthened or weakened on the basis of the information now available to the operator. The combination of static a priori and changing real-time data represents all potential information that the operator can draw upon in making judgment assessments of the tactical situation.

Clearly, there are certain physical attributes that must be present in the human component of the model for acceptable judgment performance. Although we assume the operator's physical senses are intact and performing optimally, this is certainty a simplifying assumption. A more comprehensive model would include a provision for state changes associated with the many physical parameters as well. In fact, it can easily be argued that situational awareness is conditional on the many fluctuations in the physiological state of the observer.

However, in the interest in limiting the scope of the discussion, we focus on a situational awareness system that can be defined, at least in part, through military science propositions. The basic concept of situational awareness fits rather well in a descriptive framework approach to modeling complex tactical judgment and decision making. However, there is some inherent ambiguity in the term that gives ride to multiple meanings and thus uncertainty about how it is to be conceptually defined, and how one goes about measuring it. For the present, we restrict the discussion by loosely defining the concept along the lines considered by military commanders when speaking of intelligence preparation for the battlespace. However, in this case we also consider real-time interactions with the battlespace.

In the context of decision making within a temporally bound, rapidly evolving environment, certain decision behaviors undergo modifications, as new information becomes available. There are a certain number of assumptions that comes with any military operation, and this helps set the stage for guiding a decision-maker's actions. Within this state framework conception, these fundamental assumptions are associated with 1) what is known about the tactical environment now and 2) what is known about military science constructs and propositions. These elements will tend to influence the decision making process. However, the tactical environment remains in flux to some extent, so the decision maker must always be updating what is known with respect to the events unfolding before him/her, and how this information will influence the application of certain military ideas.

It is probable that the situational state vectors, in part, would be "tuned" to the historical and doctrinal parameters, such as weather and climate, enemy force structure, and asset availability, which are typically considered during battlespace preparation. For example, in Armor systems, much of the awareness would be captured in the notion of METT-T factors (mission, enemy, terrain, troops availability, and time). Here, the situational awareness component of the state model can be conceptualized in terms of constellations of indicator variables, which occur together in well-defined patterns. While it is clear that unplanned events external to events considered during the production of orders and fragmentary orders will occur, it is unlikely that these historical events will set the thresholds for the awareness state, which itself responds moment to moment during battle.

Consider an AWACS weapons director (WD) engaged in a defense counter-air (DCA). He or she has access to various a priori historical information in the way of air tasking orders, intelligence, assets and other resources, plans and so on. This knowledge, in part, conditions the awareness state by preparing the WD to look for specific events during DCA. As the DCA evolves, the pattern of values on the dimensions making up the awareness for the tactical environment These moment to moment changes follow fluctuations in the WD attention to certain features of the DCA engagement. In this conception of situational awareness, a unique vector of values in multidimensional awareness space describes each state. Here, the vector combines the values for each of the system variables that the WD momentarily determines to be possible influences on the outcome of the engagement.

The final component of the model characterizes the action space, or the judgment alternatives available to the WD. A knowledge-based rule would link a particular awareness state configuration to the action judged best, given the situation defined by the information being attended to by the operator. For example, a particular array of values of the vectors making up the awareness state may lead to action; "send attack aircraft Delta to nearest refueling cell". However, another array of values, which presumably reflect a different tactical situation, would lead to a different action, such as vectoring a strike package to certain coordinates. At any particular time of the engagement, the conditions necessary for several mutually exclusive actions may be possible. These action for the action space for the model. While only one action is possible at any given time, a particular state space configuration can establish the necessary conditions for more than one action. That is, a given awareness configuration can map to more than one action. However, in theory these other actions will have to be deferred until that action that is judged to have the optimal outcome is completed.

The current proposal describes an effort directed at using the operational awareness concept above to create an intelligent monitoring device that can help guide experts in performing operational tasks. The key distinction to this approach over other approaches used for intellectual aiding lies in the idea that experts are knowledgeable with respect to essential military science propositions and thus the coaching system does not inform operators on these details. Instead the coach acts as a guide to assist operators with fluctuations in operational awareness that likely emerge from a number of task and individual differences factors. As such, the coaching system can provide a graded feedback intervention that is based on the level of awareness detected and the level needed for a particular tactical environment.

COACHING PROTOCOL FRAMEWORK

The intellectual coaching theory expressed in this proposal comes from the consolidation of three areas of research: 1) judgment and decision making, 2) display engineering, and 3) situational awareness. The thematic features that unite these literatures can be summarized in the 3 global scientifically validated premises below.

1) There exists variability in task properties and these properties directly constrain expert decision-making. Before one asks any questions about the nature of information processing (i.e., what is going on inside the head of the decision maker), there must be a degree of clarity concerning the characteristics of the problem and the demands that are being placed upon the

expert. This calls for a model of the decision task, or more generally- a methodology directed at modeling the ecology in which the decision-maker operates. Once this model has been identified, then one is in the position to make predictions concerning decision behaviors that are likely to occur on the basis of theoretical constructs defined within the framework of the model selected. The novelty of this approach lies in its emphasis on understanding the ecological features of the decision environment. This approach is in stark contrast with the vast majority of decision and information processing models which place nearly exclusive emphasis on the brain of the decision maker at the exclusion of the ecology in which the brain is embedded. Thus, this is an adaptive view. Task properties induce or cause particular kinds of decision behavior to occur. Understanding these properties will allow one to gauge and/or assess responses by operators to the demands of the task. Finally, understanding the response to task demands will establish the vehicle to generalize behavioral outcomes to other ecological environments. We know what to expect in the user in response to changing decision contexts.

2) There exists variability associated with the mediation of the task system. The manner in which the decision environment is expressed will induce particular information processing responses in the decision-maker. This idea is quite simple, and represents the analog form of the task property premise (#1) above. The computer interface is a window to the task world. It can be viewed as representing the surface features of a system that are linked in various ways to depth features of the real world. That is, we have knowledge of the covert properties of the world through surface information. If the window (surface information) distorts the world, responses to task demands will also be distorted.

There are many ways to construct the window and not all construction methods provide an invariant observation of the task world. Altering the size of the window may restrict or enhance information access. For example, there may be times in which the window is constructed to prevent the decision-maker from seeing certain things (e.g., irrelevant information). It can also be constructed to force the decision-maker to attend to particular aspects of the decision world. Finally, the window can be rendered intelligent in the sense that it can reconfigure itself in response to changes in task properties and changes in information processing characteristics of the decision-maker in order to assure some optimal surface-depth relationship.

3) There exists variability in the situational awareness of a decision-maker. This variance is associated with task demands (premise #1), mediation characteristics (premise #2), as well as many individual difference variables that bear on processing of complex data, such as, expertise, tolerance for risk, and propensity to engage in particular styles of cognition to name only a few. The premise here concerns the identification in the nature of this variation, and the identification of selected attributes that affect decision performance.

Dynamic Coaching Protocol (DCP)

Using the three premises outlined above, we will suggest a means to create a coaching mechanism that is executed in a manner that maintains congruence between the tripartite Task system, Mediation system, and Operational awareness system of the expert user.

Continua.

In order to monitor congruence there must a methodology that allows quantification in the states of 1)-task demands, 2) display (mediation) properties, and 3) operational awareness of the operator. This suggests the development of a set of continua (one for each system) that demarcate the range in states that can exist in a given operational context. The three dimensional state-space will contain a taxonomic cognitive principle discussed below that 1) can be used to identify fundamental cognitive awareness properties of operators that reflect the processing mode for environmental conditions and 2) that serves as the mapping mechanism that ties each dimension into a cognate integrated system.

Analysis and Intuition in Finite Space

The cognitive functions of analysis and intuition have been selected as anchor points for a construct dimension that specifies the manner in which DCP functions. The rationale behind this choice is reasonably straightforward. First, many cognitive processing characteristics can be classified as to what extent they possess analytical and intuitive properties (see Hammond, 1996). Secondly, each continuum of the DCP system can be scaled with regard to the range of analytic and intuitive properties possessed by its state indicators. For example, the task continuum can be scaled from those tasks that require analytic decomposition of decision attributes, to tasks that represent correlated information structures and must be parsed holistically. Further, the mediation continuum can also be scaled in this manner with representations that highlight the distinction of information elements creating analytic (low proximity) displays to intuitive (high proximity) displays that use object entities to transmit integral data dimensions. Finally, operational awareness can also be scaled to reflect the range of attentional tuning requirements for analytic attentional focus (elemental features of the task), to the tuning required for the intuitive attentional focus (distributed aspects of the task). Furthermore, the analysis-intuitive concept allows one to consider the level of certainty or reliably in the information system and its effect on awareness. Here, analytic precision would represent a key instantiation of a cognitive awareness mode that is manifested during a serial evaluation of deterministic decision attributes. Intuitive assessment represents the mode of activity that would match the irreducible uncertainly associated with operational tasks composed of incomplete and uncertain information, or information of limited validity. Thus, the task continuum which is mediated by the displays will define a ordered set of task properties that induce a particular cognitive awareness response in the operator, ranging from analysis to intuition.

We believe there is a method to define a "band of congruence" that quantifies the degree to which the separate systems are aligned. Deviations from this alignment state will represent potential shortfalls in operator performance. For example, through implementation of a monitoring process for deviations in congruence, adaptable coaching interventions may be automatically invoked. Thus, fluctuations in the properties of a task (e.g., say shifts from identification of radio signature of a target to a more complex threat identification, which may constitute combining multiple fallible information sources) may warrant concomitant changes in the way the coach is mediated for the operator. For example, a change from deterministic (analytic) decision requirements to one of probabilistic assessment may demand shifts from easily decomposed tabular information presentation to perceptually parsed graphically displayed information.

If one argues that the mediation techniques themselves can drive cognitive processing and awareness (premise #2), then one may be in the position to use this flexible technology to alter the awareness of a "failing decision maker, or an operator that is experiencing attentional drift. Drifting operators (operators not in congruence with task demands due to the appropriate level of operational awareness) can be brought closer to the region of optimal cognitive awareness, vis-a-vis the representational structure of the media. That is, the display will drive an operator to establish a cognitive awareness mode more closely associated with some optimal mode given particular task demands. For example, if a decision maker is in an awareness mode that is optimal for conducting an off-line assessment of a single information source, yet the task calls for a mode that complements the act of combining multiple uncertain indicators of a tactical state in real-time, then certain representational variables can be implemented to achieve this modification in awareness (discussed below).

The following sections of this paper discuss the relevant background theory that would be addressed in designing a real-time adaptable coach. In addition, the sections that follow suggest the relevant literatures that would form the basis to instantiate the notion of a layered continua model that we call Dynamic Coaching Protocol and is shown in Figure 2 below.

In Figure 2, a particular task configuration (S3 in Task State vector) will require a unique representation (S3 in Mediation State vector) that in turn activates or causes a given mode of cognition in the decision maker (S3 Cognitive State vector). This cognitive mode generates a particular course of action or decision (S3 Action Space vector). Fluctuations in alignment alter cognitive functioning (the dashed arrows emanating from the mediation vector) and cause poor performance (Error). Two Coaches, one proactive and one reactive, are used to monitor and control aspects of the mediation State vector to maintain alignment. The proactive coach analyzes archival data and identifies error inducing situations through pattern matching and frequency analysis, and the reactive coach analyzes online data to evaluate decision quality and performance in real time. As an example, suppose a decision-maker is responsible for performing a task with an S3 Task value. In the figure, if S1 in Cognitive State is activated from an S3 Mediation value, this will produce Error. The Reactive Coach presents data with an S5 Mediation value in an effort to raise the cognitive state to a value more homogenous with the S3 Task State. In addition, the intervention is archived providing data that the proactive coach can analyze and allowing it adapt to individual differences in types of errors committed .

The DCP model above illustrates two intervention pathways: 1) proactive feedforward mechanism, and 2) a reactive feedback mechanism. The Feedforward loop is viewed as important in communicating historical response information to the operator. Here, the adaptive component of the coach archives error profiles of an operator and then uses this information to trigger proactive alerts and operational awareness information to the operator. The feedback loop is triggered from an error event, which then activates the intervention toward an optimized awareness state.

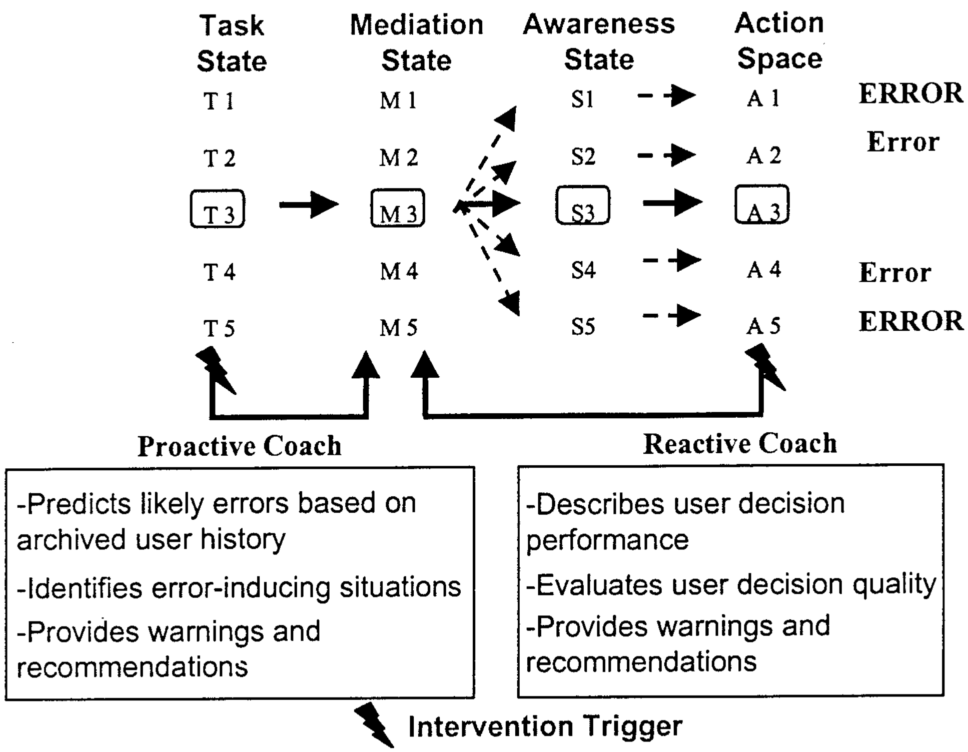


Figure 2. Layered continua model relating Task, Mediation and Awareness states to errors

COGNITIVE ENGINEERING THEORY AND APPLICATION

Cognitive Continuum Theory - Formal Model of Performance

In order to characterize the essential properties that represent intuitive and analytical forms of cognition, Hammond (1980) has elected to concentrate on principles of information organization and their precise relationship to the properties of a given task. From this organizational perspective, Hammond has developed the notion of a cognitive continuum that is anchored by analytical and intuitive modes cognition (Hammond, 1981; Hammond et al., 1987). The cognitive continuum view has allowed for the development of a theoretical framework offering specific predictions of task-driven cognitive behavior that is likely to provide a more detailed examination of human cognition over the more traditional approaches of applying normative standards in assessing cognitive efficiency. Hammond defines specific methods for testing various organizing principles, and this fact alone distinguishes the theoretical potential of the cognitive continuum from other conceptions of information organization that do not lend themselves well to operationalization and empirical verification (e.g., most notably, the notion of "schemata", Schank and Abelson, 1977; and "production systems", Newell, 1973; Anderson, 1976). Further, by predicting cognitive behavior in a manner that is independent of task definitions, Hammond avoids the circular logic in the assumption that nonanalytic processing is being manifested because an individual is performing an analytical task poorly (Garner, 1981).

A fundamental substantive contribution made by the cognitive continuum theory lies in the notion that cognitive efficiency, and thus performance, is in part, a function of the congruence between the properties of the task and the cognitive organizing principles employed by the decision maker. The theory essentially describes a system of two continua;(a) one associated with the task, and (b) the other associated with the cognitive disposition of the individual. Oversimplifying the theory (see Hammond, 1981; Hammond et al., 1987), the assertion is that various properties of the task "induce" a particular mode of cognition lying somewhere between the analytical and intuitive poles on the cognitive continuum. For example, a simple, highly structured deterministic task (e.g., simple mental arithmetic) is likely to induce a mode of cognition (i.e., organizing principle) at the analytical end of the continuum. Here, the psychological/behavioral consequences of such an organizing principle is that a very proceduralized set of operations are executed, at a somewhat methodical pace with a relatively high degree of accuracy, where the subject is highly aware of the organizing principle. In contrast, a complex, ill-structured and ambiguous task is likely to induce a mode of cognition in the person that is closer to the intuitive end of the continuum. The intuitive cognitive mode would tend to be associated with a holistic organizing principle, executed quickly and with lower overall accuracy when compared to some normative standard, and where the subject would manifest less awareness of the actual organizing principle being used in performance. In effect, the closer the congruence between the properties of the task and the optimal mode of cognition given the structure of the task, the more efficient cognition is likely to be. Thus, for example, an individual utilizing analytical skills to solve a very complex ill-defined problem will likely display incongruence between the organizing principle that is analytic in this case, and the properties of the task, which call for a very different organizing principle that is intuitive in nature. This incongruence will ultimately lead to inefficient cognition.

Tasks can also require quasi-rational organizing principles (Brunswik, 1956). Thus, a task can be defined as possessing both analytical and intuitive properties. The location of the task on the task continuum would be somewhere between the two cognitive poles. The implications for performance on such a task would be that an individual must use both analytical and intuitive skills to adequately perform the task. The nature of a quasi-rational organizing principle is consistent with other cognitive formulations for goal directed decision-making, most notably Simon's (1957) bounded rationality principle (see Hammond, 1981).

Ecological Context and Task Structure

The fundamental edifice of the cognitive continuum theory is the notion that the task is responsible for structuring (i.e., inducing) a particular form of cognition. The theory has been developed within the context of the Brunswikian notion of probabilistic functionalism and 'vicarious mediation' in visual perception (Brunswik, 1956). Brunswik's view was that visual perception is the activity characterized by a perceiver interacting with his/her ecological environment; an environment whose tendency it is to distribute or "scatter its effects". Within this context, he viewed the important ecological dimensions (or cues) of the environment as being probabilistic and not fully reliable or dependable. The fact that the environment presents the perceiver with redundant information in the form of correlated cues (i.e., the environment is vicariously mediated), means that the perceiver must wisely select and use the cues most diagnostic of a given behavioral or perceptual goal. A rather good functional example of the meaning of the probabilistic nature of environmental cues is taken from Gordon (1989) on Brunswikian Psychology:

"Suppose we are searching for an edible fruit. Let us assume that edible fruit is (a) darker, (b) redder,(c) softer and (d) sweeter. Obviously, darker and redder are visual cues, softer is tactile, sweeter is gustatory: the environment is scattering its effects. And these cues, the only ones available, are all imperfect: all carry some risk. Not all ripe fruit is red, nor is all red fruit edible. Sweetness often indicates edibility, but some poisonous fruits are sweet. Some fruit is less edible when soft, some fruit will be rotten." (pp. 131)

As a functionalist, Brunswik's basic perceptual theme was adaptive in nature. That is, in order to survive the perceiver must deal in risk and uncertainty by acting like an intuitive statistician. Thus, the perceiver must be able to (a) select meaningful cues from a plethora of ecological information, (b) factor the riskiness of the situation, (c) combine the cues and risk factors, and (d) render a judgment leading to action (e.g., avoid the thicket of trees else risk being eaten by a tiger).

Brunswik (1952, 1956, and 1957) was responsible for introducing a formal systems approach to the study of human cognition. Brunswik declared that:

"Both organism and environment each with properties of its own...... Each has surface and depth, or overt and covert regions. It follows that much as psychology must be concerned with the texture of the organism.....it must also be concerned with the texture of the environment" (1957; pp. 5)

From his probabilistic perspective on the relationship between a perceiver and his/her environment came Brunswik's lens model of behavior, which defined the structural characteristics of the person/ecology relationship. This unique model, with both normative and descriptive features, defines the complex multidimensional representation of human behavior within an ecological context (Brunswik, 1952, 1956). It has been since modified and expanded in order to represent a general model of human judgment and decision making (Tucker, 1964; Hammond and Adelman, 1976; Hammond, McClelland, and Mumpower, 1980; Brehmer and Joyce, 1988).

Figure 3 illustrates that the lens model essentially distinguishes between an object or condition that is defined by various information sources (cues), and the psychological representation of the object or condition which is defined through a particular judgment policy. The lens model portrays the environment as a series of cues whose relationships with the environment are less than perfect. A decision-maker is viewed as interacting with his or her environment through a 'lens', which is often distorted because of this imperfect and uncertain relationship. The relationship between the cues and the environment is typically characterized by "ecological validities" that, in theory, can range in absolute value from 0 to 1.0. Ecological validity represents the predictive importance of each cue. The manner in which a decision-maker uses particular cues can be modeled by a regression equation that predicts an individual's judgment of an object from a linear combination of cue weights. The degree to which a decision maker accurately assesses the characteristics of an object or condition in the environment is expressed by the correlation between the object's true values and those predicted by the decision maker (Hammond and Wascoe, 1980).

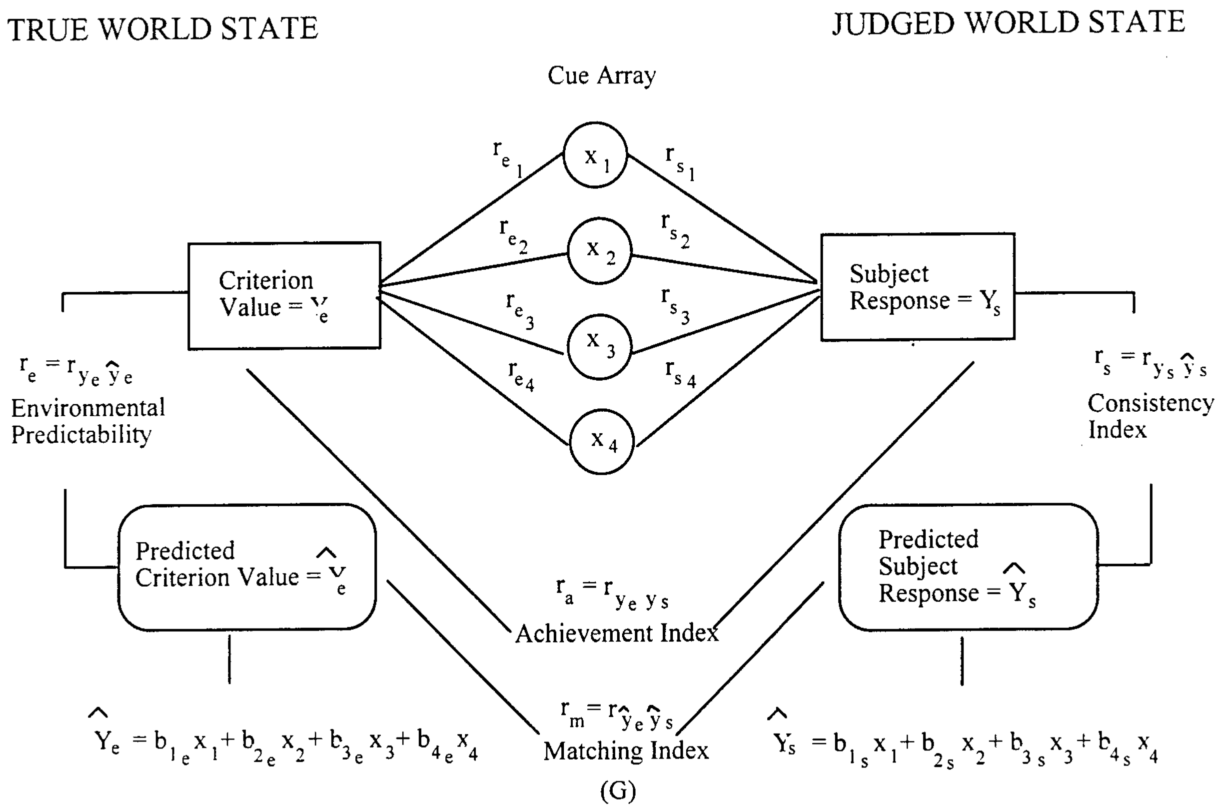


Figure 3. Formal Model of Expert Judge and Environment Interaction.

The lens model provides a formal means for quantifying the influence of various task features on human cognitive behavior. As can be seen in Figure 3, the lens model provides the means for manipulating various properties of the task, and additionally provides a network of task and cognate behavioral descriptive terms that can be useful in locating a person's cognitive activity on the continuum.

Current cognitive research efforts are responding to the importance of ecological context by acknowledging the pivotal role of task properties in cognition (i.e., what is going on outside the head). For example, in his detailed review of decision making, Payne (1982) asserted that decision making in general is very much contingent on the demands of the task. Simon (1978) also makes this point when addressing issues associated with adaptive systems by indicating that it is features of the task that strongly influence and guide overall behavior. Newell (1973) also suggests the importance of task ecology when discussing various architectural theories of cognition noting that it is environmental properties, which largely determine cognitive hardware requirements or architectural structure. Garner's (1974, 1981) highly cited work on object perception and the principles of "configurality" and "emergent features" underlying unanalyzed perception (i.e., holistic perception) bears directly on his arguments for the importance in specifying properties of the stimulus. Garner essentially argues that task properties alone serve to mediate the mode of object perception. Thus, it is task ecology that induces either a holistic mode of perception, or a perceptual strategy that is based on an analytic decomposition of the object attributes (Garner 1981).

Garner's work on object perception has led to a number of current debates in applied psychology. For example, a controversial topic is associated with the question of what kinds of data displays are actually superior at inducing holistic processing (cf., Carswell & Wickens, 1987; Sanderson, Flach, Buttigieg, & Casey, 1989). Wickens' proximity compatibility hypothesis follows from Garner's work and essentially defines the importance of compatibility between the properties of the task, and in this case, how it is presented and displayed to the operator in order to induce optimal cognitive processing. The proximity compatibility hypothesis "attempts to relate the processing of the displayed information to the nature of the task information processing characteristics" (Wickens & Andre, 1990, pp. 62).

Wickens' arguments for compatibility between display structure and cognition bears, in part, on the work documenting the search for veridical and non-veridical mental representations of physical events and attributes (i.e., how people think about the physical world they exist in) (see Gentner & Stevens, 1983 for review). The interest in this work for engineering psychology is to build displays that are most congruent with the way in which people naturally organize and use information. Moreover, the work on mental representations raises performance measurement issues of the kind Hammond et al., (1987) point out in that it seems reasonable to first understand how people think about a given phenomenon before rendering claims concerning the efficiency of human cognition that is based upon a specific performance modeling methodology.

Discussion on Implications of Cognition for Interface Design

Performance by experts <u>depends</u> on the situation in which they work, and the task requirements for performance. This perspective is echoed in the CCT framework where the task itself is viewed as being instrumental in structuring the mode of cognition used by experts in their judgments about objects, things or behavior. CCT asserts that if the task is complex, multidimensional, and possesses uncertain qualities, the most efficient approach for decision making will necessarily be intuitive in nature. If on the other hand, the task is very well defined, supporting absolute and precise statements about task parameters, an analytical mode of processing may be called for. In most cases, however, both analysis and intuition are parts of an expert's judgment. That is, tasks often call for both kinds of processing. In this case the task would induce a quasi-rational mode of cognition that lies somewhere in between the end points on the cognitive continuum. As we shall see below, the task of maintaining situational awareness in complex tactical conditions favors supporting intuitive cognition.

An additional implication of the above discussion is the notion of information format. If one accepts, at least in part, that the task is responsible for structuring and inducing a particular form of cognition, then it follows that the structure of the task must be preserved in the physical display itself. This means that the transformation from the task to the display of task parameters must be invariant so as to induce the appropriate organizing principle called for by the structural characteristics of the task. From this perspective, it is clearly possible to have a task calling for a particular cognitive mode, yet this mode is not supported by the physical properties of the display interface. This disconnect between task structure and display structure leads to ineffective cognitive performance. The physical properties of the display itself are crucial for preserving task structure.

Cognitive Engineering: DCP Delivery System

A large amount of research in perception exists on the perceptual processing of multidimensional stimuli. Historically the interest in this area has focused on the characteristics of the processing system per se. However, more recent attention has been placed on investigations in the inherent structural properties of stimuli and their effect on the perceptual process. Garner (1970) stimulated interest in this area by arguing that before one can understand the details of perceptual processing, one must understand the details associated with the structure of the stimulus.

A central issue in the research on stimulus structure has been the nature of the relation between dimensions, which represent a multidimensional stimulus. Garner (1974) has discussed two major ways in which stimulus dimensions can be related. Stimuli composed of integral dimensions produce a Euclidean metric in direct distance scaling, facilitate the discrimination of stimuli on one dimension when another dimension varies in a correlated manner, and inhibit the discrimination of stimuli on one dimension when another dimension varies in an orthogonal manner. An example of integral dimensions would be the hue and saturation of a color (Garner, 1970). Separable dimensions (e.g., separate vertical bars) produce a city-block metric in direct distance scaling and produce neither facilitation with correlated dimensions nor interference with orthogonal dimensions in a discrimination task. Psychologically, integral dimensions appear as an integrated whole, whereas separable dimensions are seen as distinct and separate. Further, dimensions tend to be integral if the existence of one dimension depends on the existence of another dimension. For example, any one of the dimensions of color (i.e., brightness, hue, and saturation) can not exist without values on the other dimensions.

The study of integral and separable dimensions has traditionally been limited to relatively simple stimuli and elementary perceptual processes. Stimuli are generally composed of physical attributes such as color or geometric form and are often limited to two binary dimensions. In discrimination and identification studies, the task is to evaluate stimuli on the basis of one dimension while the stimuli vary along another dimension. The rule relating stimuli to responses is based on physical attributes of one of the stimulus dimensions. Since these tasks are generally easy to perform, speeded responses are obtained and reaction time serves as the primary dependent variable. Classification and similarity scaling do not specify what aspects of the stimuli the subject should use in forming classes and assignment ratings. Instead, the interest is on identifying stimulus structure by the way in which the subjects globally view the stimuli. With integral dimensions, classification is based on the overall similarity structure of the stimuli; with separable dimensions, subjects group stimuli on the basis of a single dimension.

Integral dimensions appear to show an increase in the speed of processing when two dimensions are correlated and interference when the two dimensions are orthogonal (Garner, 1974; Foard & Kemler, 1984). Further, selective attention to a single dimension of an integral stimuli appears difficult to achieve because perception is dominated by the overall similarity structure, or the emergent features corresponding to holistic processing (Smith & Kemler, 1978; Garner & Felfody, 1970). In contrast, separable dimensions are those that do not show any improvement in the speed of processing when the dimensions are correlated, nor interference when they vary orthogonally. In addition, separable dimensions support focused attention on specific features of the stimulus (Garner, 1976; Pomerantz, 1981).

Taken together, the above research indicates that the structure of the stimulus plays an important role, not only in elementary perceptual operations, but also in higher order cognitive processes. Specifically, stimuli composed of integral dimensions appear to facilitate performance across several types of perceptual and cognitive tasks. Recent interest has focused on the role of integral dimensions for facilitating performance on information integration tasks.

Analysis to Intuition- Separable versus Integral Object Displays

Many recent studies have addressed the relative merits of different display formats on human performance. The results of many of those studies support the use of integrative, object-like displays for enhancing a user's ability to assimilate complex information (Carswell & Wickens, 1987; Coury et al., 1986). Such displays appear to be especially effective in situations including multidimensional task variables where decision values are intercorrelated. There is substantial evidence to suggest that object displays are superior to alphanumeric displays in many applications in which identification of system state requires integrating data from several information sources (Casey & Wickens, 1986; Wickens, 1986).

The superiority of integral displays has been attributed to the perceptual cues and redundant information inherent in such representations (see Garner, 1974). An operator can use the redundancy in perceptual cues to simplify classification of system data by associating a unique object configuration (e.g. "a happy face") to a specific system state category. Mapping objects or features to a state category occurs when the values of system variables are correlated with a particular state category and the physical representation of those values creates a configuration with a unique size, shape and orientation. Thus, the user need not attend to specific values of the system variables but can rely on rapid, perhaps holistic, integral processing of an object-like configuration or set of salient features to determine the state of the system. In terms of multiple-resource theory (Wickens, 1984), object displays produce a spatial code that allows integral processing of system data.

Alphanumeric displays and tabular information, on the other hand, require the user to attend to each individual system variable and to serially process information as a verbal code. Because these separable display formats require mental manipulation of numerical values to determine category membership, the underlying correlational structure of the system is not readily apparent (at least in Garner's 1970, 1974 sense). Presumably, because separable display formats require focusing on each system variable dimension in order to interpret and integrate across system variables, the separable display requires more processing time than the integral display. Consequently, many researchers have concluded that the appropriate display format depends upon the underlying statistical properties of data in a task (Wickens & Andre, 1990; Mahan, 1994).

Discussion on Implications of Display Design for Cognitive Functioning

The discussion on display configuration implies that there exists a very close link between the judgment process and the structural properties of the display for influencing this process. Oversimplifying, if the task is one of making judgments about a very complex system which is composed of correlated information dimensions, an integral display will be most compatible with the task at hand and will support the mode of cognition most compatible with a correlated task structure. In contrast, if the task domain is well defined and composed of orthogonal information dimensions, a separable display may be more appropriate. These ideas are especially consistent with the congruence principle offered by the Cognitive Continuum Theory.

Implementation of DCP

The discussions above suggest that displays can be created that drives an individual and team decision-makers toward a particular kind of cognition. Anderson (1991) has extensively written most recently on the adaptive qualities of cognitive functioning. Further, he notes that cognition is profoundly influenced and constrained by the properties of the environment in which the decision-maker is embedded. This position is echoed throughout the above discourse in the Brunswikian views that provide a formal approach to modeling individuals and teams in specific operational contexts.

The cognitive continuum theory (CCT) is a framework built upon the notion of the adaptive significance of human cognition. It formally defines the behavior of decision-makers within a variety of task conditions--from the well-specified and deterministic tasks that induce analytical processing, to the ill structured and uncertain tasks that induce intuitive processing. As a theory of cognition, the CCT relies upon the mathematical features of the Lens model in Figure 2 for identifying the efficacy of cognitive activity within a given task environment. CCT represents a framework for quantifying team decision making in sustained operations. Further, it allows for tracking changes in the efficacy of decision-making brought on by changes in operational task parameters.

The use of specific data packaging configurations (displays) can induce changes in processing which are themselves measurable within the CCT model perspectives, as well as using more traditional critical incidence and other frequency-based measures. Thus, real-time measures of changes in processing can serve as a stimulus triggering corresponding changes in displays, which in theory, can drive the operator toward more efficient processing efforts. Here, the physical properties of the display are used to drive the cognitive process up and down the cognitive continuum to arrive at a point on the continuum that is most congruent with task characteristics.

An important aspect of any coach is the ability to detect and categorize errors to guide future performance. Errors can differ not only in their severity (a quantitative characteristic), but also in their typology (a more qualitative characteristic). An effective coach must be sensitive to fluctuations in error severity and typology. While the severity of an error needs little explanation, the typology of an error does. For heuristic purposes, consider differences between an error made while assigning a track and identifying an enemy attack pattern. While assigning a track there is a clear available formula for how to integrate the information, you proceed at a slow, methodical rate, and your decision policy is easily retraceable (when asked how you arrived at your answer you can easily justify the means). Errors are likely due to a lack of focused attention with regard to some *specific* characteristic; for example, insufficient attention to the amount of fuel or the type armament of the track. This type of error will result if your focus is too holistic (a more intuitive mode) and will result in a commitment error. Contrast this with another type of judgment error. When judging the attack pattern of incoming bogies you should be relying on perceptual pattern recognition. Different identification patterns will result in a different action plan. This type of error can occur by focusing on the attributes of the front bogey rather than the overall pattern. In this instance, you are being analytic (focusing on specific attributes of one bogey) when the task requires a more intuitive mode (holistic evaluation of the bogey pattern). Recommendations by the coach take into account the type of error when deciding what type of feedback to provide.

DCP Response System

The absence of well-defined and precisely scaled continua means that it is difficult to quantify where, for example, a display representation is located on the proximity continuum. For example, with respect to the proximity continuum, all we can say is that one display has higher proximity than another, i.e. "more" integral or separable than another. However we cannot precisely quantify proximity as a construct. This is because proximity is tied to the psychological experience a user has in response to elements of the display (see Wickens & Carswell, 1995). Neither can we precisely quantify the cognitive continuum. For example, we can say that a user is behaving very intuitively. However, we cannot say precisely how intuitive. Thus, model-based solutions to the problem of precise quantification of task, proximity, and cognitive mode are difficult to achieve. Instead, an approach geared to the ambiguity in these continua provides for more tractable solutions in identifying regions of task properties, proximity, and cognition that are important in technological support and for the congruence construct.

Since it is difficult to base the DCP model on precise rules that define discrete points on the tripartite continua (task, display, cognitive), a fuzzy approach that defines the continua in terms of linguistic values appears particularly useful. Linguistic values obviate the need to define precise thresholds or numeric values for demarcating the ordering of states along the continua.

The first step in implementing the linguistic control feature of the DCP model is to create a rule-base that defines the possible states of the system and the membership functions that define the relationships among the linguistic values. For example, the a rule combination may look like the following:

IF task category = low AND display category = low AND operational awareness mode = low

THEN system = congruence (green cells in Figured below)

IF task category = low AND display = low AND operational awareness mode = medium

THEN system = Positive_small Deviation (red cell in figure below)

In this second condition, the DCP response is to provide the Positive_small Deviation state in order to invoke a minor change in the level of awareness and help move the user closer to a point of congruence.

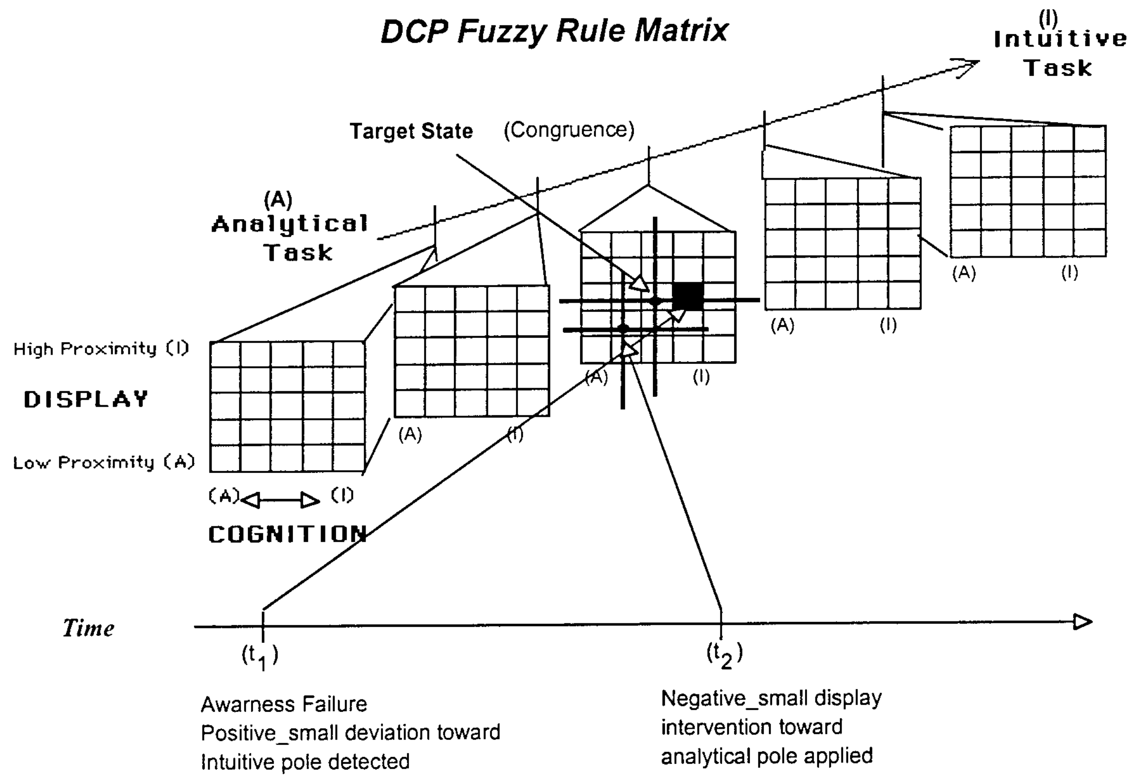


Figure 4. Dynamic Coaching Protocol rule matrix

The Fuzzy Rule Figure above graphically illustrates an overview of the rule matrix that is used to invoke display interventions in response to changes in operational awareness. Here, a fixed task state is paired with the optimal proximity-based display. The displays will range from tabular (decomposed)-numeric to closed form graphic in order to represent the separable to integral continuum on which the Analytical-Intuitive distinctions are conceived (see Wickens et al, 1995). In response to an error event or an historical marker which predicts and operator error event the operator's responses are categorized within the rule matrix and evaluated for congruence properties. When departures from congruence are detected a display intervention (changes in display format) are invoked that guide the user toward target state.

Display Components

The coaching mediation method will use both reactive and proactive display concepts. These two vehicles for delivering coaching information will provide the operational awareness responses to task specific errors and projected errors.

Figure 5 shows the coaching interface, as it might be adapted to integrate into a DDD AWACS platform framework. Included in the interface is the primary radar grid that presents two-dimensional positional information on military aircraft assets.

The Coach pull down menu is the interface to the DPS module. A number of options will be available including a switch that will activate the system to automatically monitor incoming communication and system data, as well as the responses of the WD to this information. A variety of algorithms are now under development that will measure parametric values of battlespace systems (fuel, distance to target/s, tanker locations, etc.). These algorithms will be the hearts of situational awareness mechanisms that alert the WD to potential missed or neglected information important in a given tactical scenario. Within this mode, the WD can activate speech-based or text-based natural language probes of the coach in order to understand the rational used by the coach in decision recommendations.

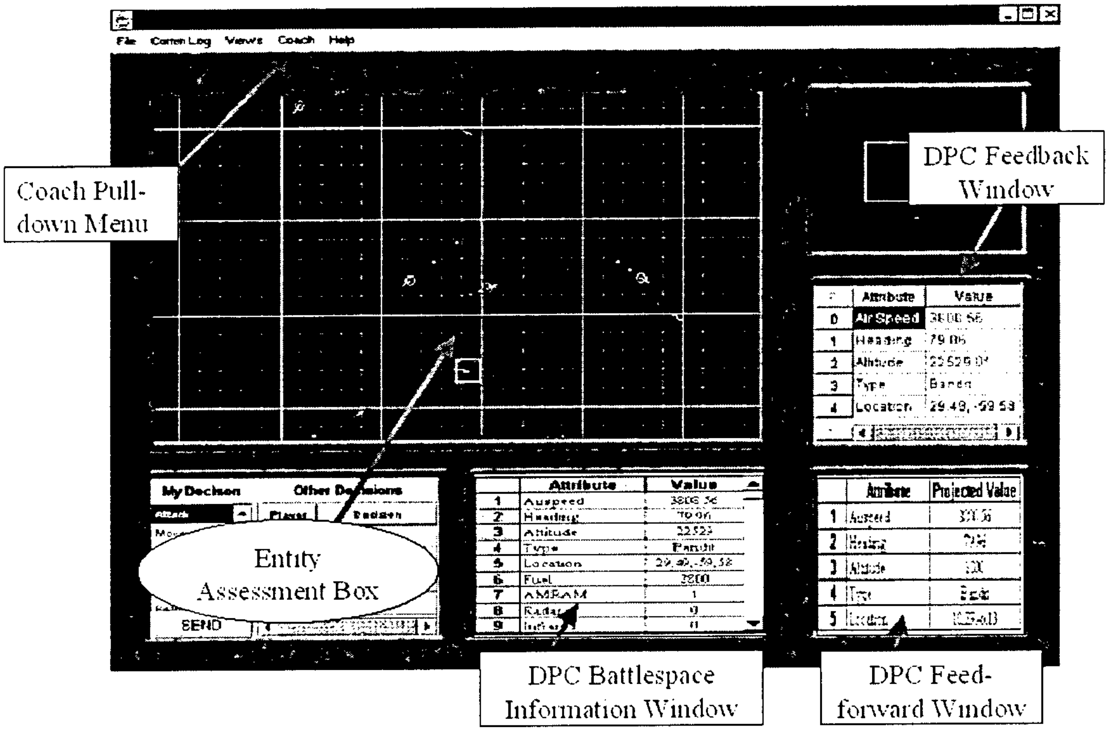


Figure 5. Coaching interface with feedback presented in tabular format

In Figure 5, an analytic-focus task state has been identified by the DPS system in response to an error linked to an uncommitted bandit nearing a high value-refueling asset. There are two immediate consequences of this detected error event. First, the DPS system highlights the bandit on the screen with an entity assessment box. Secondly, a DPS battlespace information window displays all of the critical attributes of the object that are known, such as heading, airspeed, radar type etc. This window gives overall diagnostic information regarding the object. The DPS feedback window isolates the attributes that are associated with the error event in a table of values. These attributes provide specific information on where this bandit is relative the refueling asset. The tabled values require an analytical assessment in order to precisely determine bandit's location. The Feedforward window extrapolates this information to provide a future situation-report on where the bandit will be in a given time period (configurable). The goal of the Coach is to induce in the operator an analytical assessment of this event in an effort to quantify what must be done in the immediate future. The need for attention focus in this example is due to the demand for a quantitative analysis on the bandit proximity to the refueling asset in terms of when the bandit will close with the asset. The Coach intervenes in a manner that produces a very narrow and elemental assessment of an object's precise position.

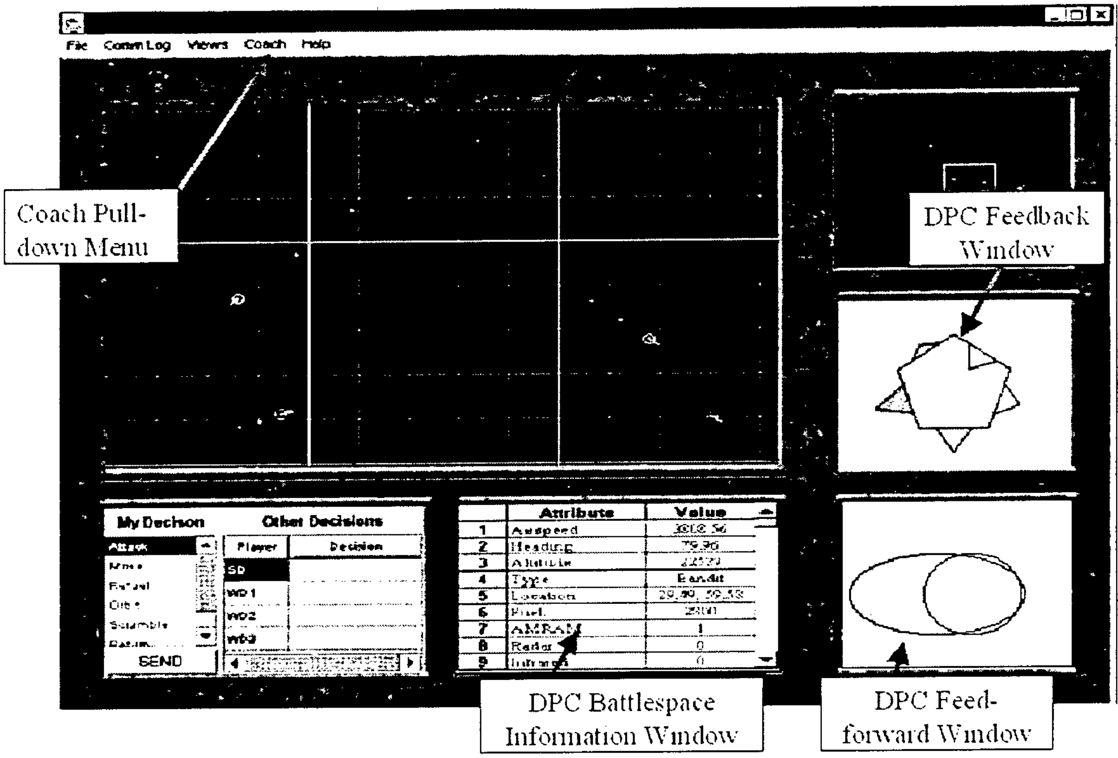


Figure 6. Coaching interface with feedback presented in graphical format

In Figure 6 above, an intuitive wide-focus task state has been identified by the DPS system in response to an error linked to a refueling alert. In this case three aspects of the DPS system have changed. First, there is no entity assessment box because this error source reflects a number of objects that must be evaluative in relation to one another and requires a broader level of operational awareness. In addition, the Feedback window has become a polygon display that maps the fuel status of several aircraft to features of the polygon. Here a fleet of aircraft has been identified as needing fuel. The polygon provides perceptual information on the relative status of the individual aircraft. Since refueling will be completed on a need basis, the polygon display provides a very rapid way in which to assess the relative need of all the aircraft without quantifying each aircraft's fuel level. The Feed-forward display provides the future state of the refueling cell. Here, an animated display shows the refueling status of a tanker. As the display changes from a circle (open refueling status) to an ellipse (closing refueling status) the operator becomes aware of the timeline to route aircraft to the tanker before the number of aircraft at the tanker exceed the timely refueling capabilities of the cell. In this scenario, the goal is to provide a coaching intervention that facilities a very rapid form of processing multiple information elements.

METHODOLOGY

A series of Five projects have been configured in an effort to validate the DCP concept as potential Coaching systems are outlined. Project number one will focus on the development of the five category state continua, as well as fuzzy algorithms that will control the DCP model. The study will represent accumulating, from the literature, the prototype task, display, and cognitive configurations that will represent the state values of each dimension. An empirical evaluation based on subjective and objective performance methods (workload measures, performance and so on) to verify that the configurations do indeed produce the orderings on the continua that has been shown to exist in other studies. The algorithms will be tested for process validity by piloting with simulated data (factual and counterfactual simulations).

Project number two will focus on establishing the congruence principle as it relates to operational awareness that which is central to the DCP concept. While, there has work cited on the principle of congruence (see Hammond, 1996 for review), there has been little work aimed at combining the three dimensions (task, display, cognitive mode) in an effort to support cognitive awareness.

Project number three will focus on making changes to the continua that are suggested from the outcomes discover in project 2. Here, state values on each dimension will be evaluated for robustness in effects on users (effect size assessment). In the case of small effect size outcomes, reconfigurations of state values will be made to produce reasonable discriminability among values.

Project Four will formally test the DCP model in a large-scale simulation venue. Results of the work will be submitted as evidence of the potential of the DACS approach to assist decision-making.

Project Five will be to rework the DCP model in light of project four outcomes. Here, modifications in rule-base for invoking display interventions will be conducted. The objective of project five is to fine-tune the DCP concept and to replicate the predicted Coaching outcomes. The project five report will culminate with the publications of the software and documentation for the DCP intervention application.

SUMMARY

The conceptual ideas advanced in this proposal underscore the notion that we have the theoretical research tools available from judgment, display engineering and operational awareness literatures to evaluate the technological coaching approach described here for enhancing the operational awareness of military personnel. Clearly, this approach is viewed as useful for getting the most out of our soldiers in a noninvasive manner. The concept of a coaching system that leverages the substantial knowledge of expert military operators by providing graded levels of feedforward and feedback information, represents a promising approach for intellectual aiding in command and control environments.

REFERENCES

Anderson, J. R. (1976). *Language, memory, and thought.* Hillsdale, NJ,: Lawrence Erlbaum.

Anderson, N. R. (1991). Decision making in the graduate selection interview: An experimental investigation. *Human Relations, 44*(4), 403-417.

Arbib, M. A. (1972). *The metaphorical brain.* New York: John Wiley & Sons.

Benbasat, I., & Todd, P. (1996). The effects of decision support and task contingencies on model formulation: A cognitive perspective. *Decision Support Systems, 17*, 241-252.

Brehmer, B., & Joyce, C. R. B. (Eds.). (1988). *Advances in psychology: Vol. 54. Human judgment: The SJT view.* North Holland: Elsevier.

Brunswik, E. (1952). *The conceptual framework of psychology.* Chicago, IL: University of Chicago Press.

Brunswik, E. (1956). *Perception and the representative design of psychological experiments.* Berkeley: University of California Press.

Brunswik, E. (1957). Scope and aspects of the cognitive problem. In H. Gruber, K. R. Hammond, & R. Jessor (Eds.), *Contemporary approaches to cognition* (pp. 5-31). Cambridge: Harvard University Press.

Bung, M. (1980). *The mind body problem.* Oxford: Pergamon Press.

Carswell, C. M., & Wickens, C. D. (1987). Information integration and the object display: An integration of task demands and display superiority. *Ergonomics, 30*, 511-527.

Casey, E. J., & Wickens, C. D. (1986). *Visual display representation of multidimensional systems* (Tech Report CPL-86-2/MDA903-83-K-0255). Champaign: University of Illinois Cognitive Psychology Lab.

Chu, R. W., Mitchell, C. M., & Jones, P. M. (1995). Using the operator function model and OFMspert as the basis for an intelligent tutoring system: Toward a tutor/aid paradigm for operators of supervisory control systems. *IEEE Transactions on Systems, Man, and Cybernetics, 25*, 1054-1075.

Coury, B. G., & Drury, C. G. (1986). The effects of pacing on complex decision-making inspection performance. *Ergonomics, 29*(4), 489-508.

Crynes, B. L., & Hawley, J. A. (1995). Professor silicon and professor maestro-The perfect combination. .

Fleming, J. L., & Horwitz, C. (1996). *Applications of the Rapid Intelligent Tutoring System Development Shell (RIDES).* Paper presented at the ITS workshop on architectures and methods for designing cost-effective and reusable ITSs.

Foard, C. F., & Kemler, D. G. (1984). Holistic and analytic modes of processing: The multiple determinants of perceptual analysis. *Journal of Experimental Psychology: General, 113*(1), 94-111.

Garner, W. R. (1970). The stimulus in information processing. *American Psychologist, 25*, 350-358.

Garner, W. R. (1974). *The processing of information and structure*. Hillsdale, NJ: Erlbaum.

Garner, W. R. (1976). Interaction of stimulus dimensions in concept and choice processes. *Cognitive Psychology, 8*(1), 98-123.

Garner, W. R. (1981). The role of configuration in the identification of visually degraded words. *Memory & Cognition, 9*(5), 445-452.

Garner, W. R., & Felfoldy, G. L. (1970). Integrality of stimulus dimensions in various types of information processing. *Cognitive Psychology, 1*, 225-241.

Gentner, D., & Stevens, A. L. (1983). *Mental models*. Hillsdale, N.J.: Lawrence Erlbaum Associates.

Glover, S. M., Prawitt, D. F., Spilker, B. C., & O. (1997). The influence of decision aids on user behavior: Implications for knowledge acquisition and inappropriate reliance. *Organizational Behavior and Human Decision Processes, 72*, 232-255.

Gonzales, A. J., & Ingraham, L. R. (1994). Automated exercise progression in simulation-based training. *IEEE Transactions on Systems, Man, and Cybernetics*(24), 863-874.

Gordon, I. E. (1989). *Theories of visual perception*. Chichester: Wiley.

Haarbauer, E., Dunwoody, P. T., Marino, C. J., & Mahan, R. P. (Submitted for publication) The team as organism and environment: Developing Brunswikian Perspectives of team modeling.

Hammond, K. R. (1980). *The integration of research in judgment and decision theory* (Report No. 226). Boulder, CO: Center for Research on Judgment and Policy, University of Colorado.

Hammond, K. R. (1981). *Principles of organization in intuitive and analytical cognition* (Report No. 231). Boulder, CO: Center for Research on Judgment and Policy, University of Colorado.

Hammond, K. R. (1996). *Human judgment and social policy: Irreducible uncertainty, inevitable error, unavoidable injustice*. New York: Oxford University Press.

Hammond, K. R., & Adelman, L. (1976). Science, values, and human judgment: Integration of facts and values requires the scientific study of human judgment. *Science, 194*(4263), 389-396.

Hammond, K. R., Hamm, R. M., Grassia, J., & Pearson, T. (1987). Direct comparison of the efficacy of intuitive and analytical cognition in expert judgment. *IEEE Transactions on Systems, Man, and Cybernetics, SMC-17*, 753-770.

Hammond, K. R., McClelland, G. H., & Mumpower, J. (1980). *Human judgment and decision making: Theories, methods, and procedures*. New York: Praeger.

Hammond, K. R., & Wascoe, N. E. (1980). *Realizations of Brunswik's Representative Design*. San Francisco: Jossey-Bass Inc.

Hatano, G., & Inataki, K. (1992). Desituating cognition through the construction of conceptual knowledge. In P. Light & G. Butterworth (Eds.), *Context and Cognition: Ways of Learning and Knowing* (pp. 115-133). New York: Harvester Wheatsheaf.

Janis, I. L. (1972). *Victims of Groupthink.* Boston: Houghton Mifflin.

Kontogiannis, T. (1996). Stress and operator decision making in coping with emergencies. *International Journal of Human-Computer Studies, 45*, 75-104.

Lesgold, A., Eggan, G., Katz, S., & Rao, G. (1992). Possibilities for assessment using computer-based apprenticeship environments. In W. Regian & V. J. Shute (Eds.), *Cognitive Approaches to Automated Instruction* (pp. 49-80). Hillsdale, NJ: Lawrence Erlbaum Associates.

Mahan, R. P. (1994). Stress induced strategy shifts toward intuitive cognition: A cognitive continuum framework approach. *Human Performance, 7*(2), 85-118.

Mumaw, R. J., & Roth, E. M. (1992). *How to be more devious with a training simulator: Redefining scenarios to emphasize cognitively difficult situations.* Paper presented at the Simulation Multi-Conference: Nuclear Power Plant Simulation and Simulators, Orlando, FL.

Newell, A. (1973). Production systems: Models of control structures. In W. G. Chase (Ed.), *Visual information processing* (pp. 283-308). New York: Academic.

Ozturk, P., & Aamodt, A. (1998). A context model for knowledge-intensive case-based reasoning. *International Journal of Human-Computer Studies, 48*, 331-355.

Payne, J. W. (1982). Contingent decision behavior. *Psychological Bulletin, 92*(2), 382-402.

Pomerantz, J. R. (1981). Perceptual organization in information processing. In K. M. & J. R. Pomerantz (Eds.), *Perceptual Organization* (pp. 141-180). Hillsdale, NJ: Erlbaum.

Rush, R., & Wallace, W. A. (1997). Elicitation of knowledge from multiple experts using network inference. *IEEE Transactions on Knowledge and Data Engineering, 9*, 688-696.

Sanderson, P. M., Flach, J. M., Buttigieg, M. A., & Casey, E. J. (1988). *Object displays do not always support better failure detection in integrated tasks* (Working Paper EPRL-88-06): Urbana: University of Illinois, Engineering Psychology Research Laboratory, Department of Mechanical and Industrial Engineering.

Schank, R. C., & Abelson, R. P. (1977). *Scripts, plans, goals and understanding: An inquiry into human knowledge structures*. Hillsdale, NJ: Lawrence Erlbaum.

Schofield, J. W., Eurich-Fulcer, R., & Britt, C. L. (1994). Teachers, computer tutors, and teaching: The artificially intelligent tutor as an agent for classroom change. *American Educational Research Journal, 31*, 579-607.

Silverman, B. G. (1995). Knowledge-based systems and the decision sciences. *Interfaces, 25*(67-82).

Simon, H. A. (1957). *Models of man: social and rational; mathematical essays on rational human behavior in a society setting.* New York, New York: Wiley.

Simon, H. A. (1978). Rationality as a process and product of thought. *American Economic Review, 68*(1-16).

Smith, L. B., & Kemler, D. G. (1978). Levels of experienced dimensionality in children and adults. *Cognitive Psychology, 10*(4), 502-532.

Srisethanil, C., & Baker, N. (1995). ITS-Engineering: Providing adaptive teaching in the engineering tutor. .

Todd, P., & Benbasat, I. (1994a). The influence of decision aids of choice strategies: Under conditions of high cognitive load. *IEEE Transactions on Systems, Man, and Cybernetics, 24*, 537-547.

Todd, P., & Benbasat, I. (1994b). The influence of decision aids on choice strategies: An experimental analysis of the role of cognitive effort. *Organizational Behavior and Human Decision Processes, 60*, 36-74.

Tucker, L. R. (1964). A suggested alternative formulation in the developments by Hursch, Hammond, and Hursch and by Hammond Hursch and Todd. *Psychological Review, 71*, 528-530.

Turner, R. M. (1998). Context-mediated behavior for intelligent agents. *International Journal of Human-Computer Studies, 48*, 307-330.

Wickens, C. D. (1984). *Engineering psychology and human performance.* Glenview, Illinois: Scott, Foresman and Company.

Wickens, C. D. (1986). *The object display: Principles and a review of experimental findings* (Tech. Report CPL-86-6). Champaign: University of Illinois, Cognitive Psychology Laboratory.

Wickens, C. D., & Andre, A. D. (1990). Proximity compatibility and information display: Effects of color, space, and objectness of information integration. *Human Factors, 32*, 61-77.

Wickens, C. D., & Carswell, C. M. (1995). The Proximity compatibility principle: Its psychological foundation and relevance to display design. *Human Factors, 37*(3), 473-494.

Appendix II:

DDD-JavaCoach Software Users Guide

How to Install and Run the Java Coach:

Follow these steps to run the Java Coach along with the DDD from the Linux environment.

1. Create a directory called "JAVA" in the user's home directory.

2. Install the jdk: Copy the following tar.gz files into the JAVA directory:

```
i18n117_v1a.tar.gz
jdk117_v1a.tar.gz
jre117_v1a.tar.gz
swing-1.0.3.tar.gz
xml4j_2_0_6.tar.gz
```

To install the jdk, merely untar each of these files using the command

```
tar -xzvf  [unique name part].tar.gz
```

This will create directories within the JAVA directory.

3. Copy the JavaCoach.jar file into the JAVA directory. Whenever a new JavaCoach.jar file is developed, you can just copy it in without any modifications needed before you run it.

4. Add the following lines in the ".login" file of the user's home directory. Be sure to substitute the correct path to the user's home directory for "/home/harrison":

```
#-----------------------------------------------------------------
# Java stuff to run coach client program.


setenv PATH "${PATH}:/home/harrison/JAVA/jdk117_v1a/bin"
setenv PATH "${PATH}:/home/harrison/JAVA/jdk117_v1a/lib/classes.zip"

setenv CLASSPATH
.:/home/harrison/JAVA/JavaCoach.jar:/home/harrison/JAVA/swing-
1.0.3/swingall.jar:/home/harrison/JAVA/xml4j_2_0_6/xml4j.jar

#-----------------------------------------------------------------
```

5. After you have logged out and logged back in, or run "`source .login`", make sure that there is no other version of the jdk running that is superceding your version. To do this, type

```
which java
```

You should get back the path to the jdk you just installed.

6. Make sure you're using the JavaCoach version of the DDD. Do this by typing

```
useddd coach
```

And then recompile the DDD by typing

```
remake
```

DDD Modules written for Coach (Programmer's Manual)

Here are the changes to the DDD to accommodate COACH that a DDD programmer would be interested in:

There are two C source code files and three header files related to the Java Coach. The two files "coach.c" and "coach_checks.c" are both in DDD/src/local_lib. The general header file "coach.h" is in /src/include with the other general DDD include files. The function prototype files "coach_fp.h" and "coach_checks_fp.h" are in the /src/include/function_prototypes directory with the other DDD function prototype header files.
Most of the externally called functions are in coach_checks.c. The initialization routines for the coach structures are in coach.c. These two files could be combined into one long file to encapsulate the coach further. The coach structures themselves are in include/coach.h. Within the coach*.c files, functions are arranged in order so that functions only call functions defined above them within the file.

One change was made to the DDD itself to get the ICG (intercept geometry) checking part of the coach to work. This change is commented in the files affected: select_icg.c and pursue.c. Prior to this change, all geometries were changed in select_icg.c to ICG_PURSUIT, since the difference between the available ICGs are not yet implemented in the DDD. In this new version, this change is not made until pursue() is called, in order for the check for appropriate ICG choice to be put into pursue(). It was not feasible to put the check in select_icg().

coach_checks.c - (Interface with DDD) Logic for triggering coach events

Throughout the rest of the DDD, all source code related to the Java Coach is contained within "#if JAVACOACH ....#endif" compilation bracketing. For the most part, these consist of single calls to functions in coach_checks.c. Attempts were made to have as little as possible relating to the Coach in the rest of the DDD. For example, if a Java Coach event was to be triggered only if a parameter was set, that parameter was passed to function in coach_checks.c to perform the check, so that within the #if JAVACOACH...#endif there is only a single call to a function, not enclosed in a conditional.

Another item of possible interest is a macro to find the distance between two points that is at the top of coach_checks.c. There are many places in the DDD code where the distance between two points is calculated, but no utility function exists in the DDD to do this calculation. In case the reason for this deficiency was concern at slowing the real-time code by adding another function call to the stack, the distance calculation in coach_checks.c was implemented as a macro, not a function. I am not certain as to the utility in moving the macro to a more general header file. Were the macro no longer at the top of the C-source code file in which it is used, it would be difficult to check the order of the x1, x2, y1, y2 parameters to be input to the macro.

coach.c and coach.h- Data Structures, Quasi-XML, Unix Connection

The source code file coach.c can be seen as being in three sections: initialization routines for the Coach Events declared in coach.h, routines for generating the quasi-XML used to communicate with the Coach, and routines containing the UNIX functions used to handle the network connection to the Java Coach process.

Coach Event Structures

The declarations and functions at the top of the coach.c file are initialization routines for the coach event data structures declared in the coach.h header file. The structure type CoachEvent, declared in coach.h is a container for all the types of events to be communicated from the DDD to the Java Coach. The fields of CoachEvent contain all the items that are communicated in every sort of message, such as the pace of the game, the DDD time in seconds, and a numeric error flag, whose meaning depends on the type of the event. The specific, different event types are declared in separate structure types in coach.h, and accessed from a union in CoachEvent. Were this all to be implemented using classes, the CoachEvent would be the parent class and the other specific event types would extend or inherit from it. The initialization routines at the top of coach.c would be the class constructors contained in these different classes.

```
   //------------------------------------------------------------------
   // Use the CoachEventStruct to approximate the base class for these
events.
   // Were the above events classes, they would all inherit from CoachEvent

   typedef struct CoachEventStruct {
       int type;    // code for type of event in uval
       int id_num;
       int pace;
       int time;    // DDD time, in seconds
       int error_flag;  // 0 if no error; nonzero values depending on type to
signal nature of error
       union {
         struct InterveneEventStruct intervene;
         struct CommitFuelCheckEventStruct      commitFuelCheck;
         struct CommitROECheckEventStruct       commitROECheck;
         struct CommitICGCheckEventStruct       commitICGCheck;
         struct EgressCheckEventStruct          egressCheck;
         struct CAPCheckEventStruct             capCheck;
         struct GroupCommittedCheckEventStruct  groupCommittedCheck;
       } uval;
     } CoachEvent, *CoachEventPtr;

   //------------------------------------------------------------------
```

Functions to generate Quasi-XML

Messages are sent to the Java Coach using a syntax that can be considered as a subset of the XML language. Eventually, a validating parser for a DDD/Coach-related grammar for XML could be developed. For now, the event types are passed using the tag <JavaBean CLASS=...>, and their field values are all passed as properties. For example, information on a Commit Egress Check Event (that is, an exiting asset entering a forbidden zone) would be passed as follows:

```
<JavaBean CLASS="org.ahrp.coach.CommitEgressCheckEvent">
  <Properties>
    <Property NAME="EventID">5</Property>
```

```
    <Property NAME="Pace">1</Property>
    <Property NAME="Time">346</Property>
    <Property NAME="ErrorFlag">1</Property>
    <Property NAME="AssetID">13</Property>
  </Properties>
</JavaBean>
```

Indentation:

A function called indent() is called by other functions that are generating the XML. The purpose of indent() is to indent the appropriate number of spaces, according to the amount of nesting of XML tags in which the current line of XML occurs. This level of nesting is tracked using the variable "level" which is declared in the main xml-generating function make_xml_string() and passed to every function that starts or ends a nesting level. The make_xml_string() function adds or subtracts from the level according to whether it is about to call a function to print a beginning or an ending tag.

At some point it was decided to not send to the Java Coach XML that had the logical-level indentation, but just to start every line indented by one space. In order to preserve the option of going back to indenting according to nesting level, the constant ONESPACE was defined, just above the indent() function. Within the function indent(), there is a compilation switch that checks if the constant ONESPACE is defined. If so, then indent() simply returns the value 1, whatever the level of logical nesting of the XML. If the programmer wishes to reinstate logical-level indentation, merely comment out the line in which ONESPACE is defined. Logical-level indentation is not currently necessary for the XML parser used by the Java Coach, but it is convenient for a human reader.

To change the number of spaces that are indented per logical level, change the value defined for the constant INDENT_PER_LEVEL in the header file coach.h. It is currently set to 2 spaces per logical nesting level.

Unix Network Connection between DDD Local Process and its Java Coach

The connection between the DDD and the Java Coach processes is set up by a function in the coach.c file called coach_connect(). This function opens a port to listen for a signal from the Java coach program, and then forks off a process that starts up the Java coach program. If you look in the coach.c file for the comment "Routines to Communicate with Java Coach Program",.you will see this and other functions that handle the interprocess communication.

Steps for adding a new event to the quasi-XML:

To add a new event to be passed to and parsed by the Java Coach, you must have agreement with those implementing the Java Coach as to the exact information to be passed for the new event, and what conditions will trigger it. You must verify that you can place a check for these conditions in some part of the DDD.

There are the specific minimum changes that must be made in coach.h, coach.c. Changes to be made in coach_checks.c and elsewhere in the DDD are less specific as they are more dependent on the circumstance.

The following summary in the comments in coach.h is a useful guide:

```
   // To define a new event type:
   //      Give it a type number using #define, and declare a struct for it.
   //      Add it to the union in CoachEventStruct
   //      Create an initialization routine in init_events(), that sets type
field to new #.
   //      And put case for it in both switch/case parts of make_xml_string()
```

Here are the steps explained in more detail:

*Changes in **coach.h:***

1) Define a new event number to come at the end of the current list of events>

```
   #define COACH_EVENT                    0
   #define INTERVENE_EVENT                1
   #define COMMIT_FUEL_CHECK_EVENT        2
   #define COMMIT_ROE_CHECK_EVENT         3
   #define COMMIT_ICG_CHECK_EVENT         4
   #define COMMIT_EGRESS_CHECK_EVENT      5
   #define COMMIT_CAP_CHECK_EVENT         6
   #define GROUP_COMMITTED_CHECK_EVENT 7
→#define NEW_EVENT  8
```

2. Create a structure for the new event. The fields of this structure should include any information other than the id_num, pace, time and error_flag that need to be sent to the Java Coach about this event.

```
   typedef struct NewCheckEventStruct {
     int first__int_thing;
     int second_int_thing;
     char string_info_for_this_new_event[NEW_EVENT_STRING_SIZE]
   } NewCheckEvent, *NewCheckEventPtr;
```

3. Define any parameters you might need, for example for an array or string size.
#define NEW_EVENT_STRING_SIZE 124

4. Add the new structure type to the union uval in the CoachEvent structure type:

```
   typedef struct CoachEventStruct {
       int type;
       int id_num;
       int pace;
       int time;     // DDD time, in seconds
       int error_flag;  // 0 if no error; nonzero values
depending on type to signal nature of error
       union {
         struct InterveneEventStruct intervene;
         struct CommitFuelCheckEventStruct
commitFuelCheck;
```

```
        struct CommitROECheckEventStruct           commitROECheck;
        struct CommitICGCheckEventStruct           commitICGCheck;
        struct EgressCheckEventStruct              egressCheck;
        struct CAPCheckEventStruct                 capCheck;
        struct GroupCommittedCheckEventStruct
groupCommittedCheck;
  →     struct NewCheckEventStruct                 newCheck;  ←
      } uval;
  } CoachEvent, *CoachEventPtr;
```

5. If the new event needs more explicit error codes than NOT_ERROR (no error situation) or COACH_ERROR (there is an error situation), then define them. For example, these were the error codes defined for intercept geometry.

```
  // For ICG Intercept Geometry errors:
  #define IR_CUTOFF     1     // IR missiles but cut-off geometry
  #define RADAR_STERN   2     // radar missiles, but stern or
stern/convergence
  #define NO_WEAPONS    3     // The plane has no weapons
remaining
```

Note that the first of these explicit error situation codes, IR_CUTOFF is given the same value as the general COACH_ERROR. There is no conflict in this, since the NOT_ERROR flag is not overwritten with an error situation.

*Changes in **coach.c:***

6. Write an initialization routine for your structure. It should take as its single parameter a pointer to a general CoachEvent. Its first line should be a call to initCoachEvent(). (This is similar to how a Java constructor would call automatically its parent constructor as its first action.) The next lines should initialize all the fields in your structure that you newly defined in coach.h.

7. In the function make_xml_string(), wherever there is a switch on Coach Event type

```
        switch (thing->type) {
```

you should add a case statement for your new type of event.

```
        case NEW_CHECK_EVENT :
```

Make sure that the names of your "bean"-event and those of its properties that you are putting into the XML string in this function exactly match the names that the Java Coach is expecting.

8. Add any other changes to make_xml_string that the logic of your new event requires. The functions above make_xml_string() are all helper functions for creating the XML and should be used. You can model your additions on how previous events were implemented in the code.

*Changes in **coach_checks.c** and **the rest of the DDD**:*

9. Write routine(s) in coach_checks.c to check for the conditions that should trigger the new event. Call the new check routine from the appropriate place(s) in the DDD. Make sure that the call to the new coach_check routine is enclosed within #if JAVACOACH....#endif.

Appendix III:
INTACT COMMERCIALIZATION STRATEGY

Overview

A growing market: According to Business Week (April 19, 1999), the market for computer-based and network-based education and training products and services is projected to be $90 billion in 2002. The market for on-the-job training products alone represents more than 12% of this projected amount. The technology for the embedded C2 coaching tool (INTACT) that Aptima proposes to develop under this STTR project has the potential to tap this extraordinary new market. Although the content and coaching strategies of INTACT are tailored for AFOSR military customers in Phase II, its sound underlying pedagogical principles, basic network technology, and unique adaptive construct make it an ideal candidate for commercialization in several technology/market pairs in the general training market.

A commitment to transition and commercialization: Daniel Serfaty, Aptima's principal founder and Chief Scientist, has a solid record of marketing and spinning off the technological ideas, methods, and products emerging from the R&D activities at Aptima. He is personally committed to apply his entrepreneurial skills to lead the project team in pursuing an aggressive commercialization strategy for the INTACT product from the early stages of Phase II. For example, working from a current SBIR Phase II project for AFRL, Aptima has been able to add almost $500K in supplementary funds from the Air Force and other customers interested in supporting the technology, as well as an additional $500K directly from the AWACS project manager's office. This vision, combining technical excellence with a focused marketing approach, is what has enabled Aptima to grow exponentially over the last three years into a $2.5M/year human-centered engineering company for both commercial and government customers. Aptima's partner in the proposed effort, Prof. Robert Mahan, Director of the Advanced Human Resource Project (AHRP) Laboratory at the University of Georgia, will collaborate in developing and implementing this strategy starting in Phase II of this SBIR project and continuing toward Phase III transition and commercialization. As head of the AHRP, Dr. Mahan has proven his ability to transition basic research principles to funded Government programs as demonstrated in the fact that the lab itself is fully funded by the Air Force Office of Scientific Research and the Army Research Institute.

A specified roadmap to success: Our approach to commercialization is pursue opportunities within both the Government and private sectors. Our ability to approach commercialization from two parallel tracks is based on building our coaching tool on a strong theory-based foundation. Too many training and decision support tools have failed in the past because they have been developed ad-hoc, without general underlying instructional and pedagogical principles. Our theory-based approach allows the proposed coaching methods to be platform independent, focused on performance enhancement and training, and thereby applicable to a variety of domains. Armed with this dual-use strategy, we plan to build on our Phase II effort and follow a clearly defined development track within the AWACS community. We will first leverage this work into one or more alternative Government application (i.e., UCAV Command and Control, Navy Aegis and Air Traffic Control) and ultimately transition into commercial domains.

What is the first product that this technology will go into?

The key technology developed under this SBIR, the Intelligent Tactical Coaching Tool, or INTACT, has been prototyped in Phase I, and will be further developed in Phase II, as an embedded tool to support operational performance and training in command and control organizations. We have designed our prototype for use on the AWACS platform and we believe that our tool can play a key role in ongoing command and control re-engineering projects. We have laid out a programmatic development approach that transitions INTACT from research tool to a real-world embedded performance support tool. This plan is presented in Table 1. As can be seen, we will work with the Air Force's research organizations to validate our tool and introduce it to the operational community through its training organizations. This approach will build credibility and buy-in from the operational community and will ease transition on the AWACS aircraft itself.

Table 1. Programmatic Development Approach for AWACS-Based INTACT

Application	Justification	Outcome
C3STARS, AFRL (Brooks AFB and Mesa, AZ)	Centers of excellence for AWACS C2 research	Conduct human-in-the-loop experimentation to validate and refine coaching tool
Tyndall AFB	Initial training center for weapons directors, air battle managers, and ground radar controllers	Demonstrate value of embedded coach for initial training as a supplement to academic training programs
Tinker AFB	Home of the AWACS Wing and advanced simulator-based training programs	Illustrate coach ability to support advanced mission and tactical training, as well as provide real-time mission support
Situation Display Console on E-3 Aircraft	Ultimate outcome of development project is to create tool to support the warfighter during missions	Prove mission utility to real-time decision support and significance of a "train like we fight" approach to AWACS operator development

Who will be your customers and what is your estimate of the market size?

The uniqueness of the tool comes from its reliance on a comprehensive model of intelligent coaching. Although the need for such a tool was initially expressed to support training and mission performance of military command and control organizations, it is equally applicable to non-military organizations. The technology underlying INTACT is generalizable to products in support of other Government, industrial, and civilian applications. Real-time, embedded decision support is directly applicable to numerous domains and we intend to target INTACT to both government and commercial customers.

Several transition opportunities have been identified for government customers. Three examples, the Navy, the FAA and the Coast Guard, are presented below, with specific potential Phase III sponsors:

- *Navy ATD for Advanced Embedded Training Systems*: This established Navy program is intended to design training and decision support tools for Aegis Battle Staff. This program also presents an opportunity to transition into commercial application by working with Lockheed Martin's Advanced Technology Laboratory. Their mission is similar to the ATD in that they are building tools to support the next generation of battle staff for the DD-21—the destroyer of the 21st century—currently in the concept development phase. A primary goal of DD-21 is to conduct combat operations with a smaller battle staff, without requiring excessive training. INTACT and the INTACT design process can aid in the development of these support and training systems.

- *Air Traffic Control (ATC)*: INTACT can be used to support current ATC operations by advancing the training process and providing real-time decision support. A more critical FAA need may be dealing with the free fight environment. Most would agree that free flight will fundamentally change the way ATC is managed, requiring new methods of training and increasing the need for embedded performance support. Potential sponsors include the Volpe Research Center in Cambridge, Massachusetts, the FAA Technical Center, and the Air Traffic Surveillance Group at Lincoln Laboratory. Aptima has already identified contacts in all these organizations.

- *U.S. Coast Guard*: INTACT can play dual roles for the Coast Guard. Internally, Coast Guard search and rescue, and drug interdiction operations can be supported by embedded performance support, specifically the command and control aspects of these tasks. Externally, we can work with the Coast Guard to develop training tools for commercial maritime operations focusing on understanding and following the "rules of the road." A potential sponsor for these efforts is the USCG Research and Development Center in Groton, Connecticut.

In addition to these Government opportunities, we are committed to exploring commercialization opportunities in the private sector. We recognize that prior to selecting a specific strategy, Aptima needs to use its marketing and business development resources to zero-in on particular technology/market pairs to start its commercialization approach. For example, the primary education market may be more interested in an assisted intelligent coach (i.e., with human teacher present) than the corporate training market. Selecting the right application for the right market is a key ingredient to our approach. Doing it differently would be extremely costly and probably fatal to the project. As a start, Table 2 provides a summary of potential applications and how we intend to approach commercialization.

Table 2: Adapting Commercialization Strategy to Technology/Market Pairs

Application Domain	***Market/Customer***	***Commercialization Potential Strategy***
Educational Software - Self-guided study - SAT preparation	Publishing Houses Board of Education	Team up with a publisher to adapt prototype Licensing technology
Corporate Decision Making - Management training - Project management	Large Corporations Training Organization	Develop coaches for business games (scaled worlds) Intranet-based licensing
Driver Education - Embedded simulation	Department of Motor Vehicles Driver Education Programs	Partner with driver simulation company
Smart Home Appliances	Home Appliances Manufacturers (e.g., GE)	Sell INTACT software module to optimize use of networked appliances by naive users
Money Management - Bookkeeping / accounting - Investment strategies	Accounting Firms Finance Software Developers	Develop add-ons for intelligent help on financial software products (e.g., Quicken)

How much money will you need to bring the technology to market and how will you raise the money?

We currently estimate that the Phase II funding, if approved, will provide an opportunity to turn the prototype into an Alpha version product. Aptima and U. Ga. are committed to raise an additional $250K from non-SBIR sources to extend the functionality, marketability, and technical capabilities of the INTACT product. In the first 12 months of a current SBIR Phase II for the Air Force, Aptima has already raised, through intensive marketing efforts, non-SBIR Phase II matching funds of $400K, with the firm promise of at least an additional $600K in the next 12 months. In addition, we successfully marketed this project to an Air Force Program Office and secured $500K in Phase III funds. We firmly intend to pursue a similar approach for this STTR project. In addition, Aptima will pursue an active partnership (i.e., cost-sharing) with a larger private sector company, such as GE Appliances Division, to help finance the implementation of a marketing, positioning, and distribution strategy, aimed at commercializing the INTACT products.

Does your company contain marketing expertise and if not, how do you intend to bring that expertise into your company?

The Aptima team has strong marketing credentials. Daniel Serfaty, Aptima's founder, has an MBA in international management, with an emphasis on industrial marketing, and an exceptional business development record in the last 15 years. At Aptima, he will get marketing support from

Meg Clancy (MBA) who has more than 18 years of experience in the management and marketing of technology. Robert Mahan, director of the AHRP Laboratory has successfully marketed basic research principles to funded Government programs at both the Air Force Office of Scientific Research and the Army Research Institute.

In addition, in the past few months, Aptima has retained the services of USA Marketing, Inc., a New England firm specializing in marketing communications for small high technology and software companies. USA Marketing is implementing Aptima's strategic vision of identifying opportunities to commercialize its human-centered technology products and services. Aptima will use its internal funds to task USA Marketing to develop a marketing campaign concept for the INTACT product, which will be then coordinated and implemented by Aptima and its partners.

Thus far, Aptima has been awarded two Phase II SBIRs and one Phase III (see commercialization report). To date, our marketing and transition funding (Phase III) has exceeded all expectations. We have already secured about $400K of external funds and are well on our way to pass the $1M mark before the first year of the Phase II is over. This represents more than a 1:1 matching ratio.

Who are your competitors, and what is your price and/or quality advantage over your competitors?

For a product like INTACT, Aptima's competitors are very fragmented by application domain, rather than by technology. For example, intelligent tutoring systems (ITS) technology has had rather limited applications, mostly to support high school scientific learning (e.g., algebra, biology). Their failure to successfully penetrate other domains (e.g., aviation, maintenance, troubleshooting) stems from the fact that very few studies of the potential users of the technology were conducted *prior* to technology development ("you cannot train or support what you do not understand.")

One of Aptima's strengths, and associated competitive advantages, is in understanding thoroughly the potential users of our products, whether they are novices or experts. We will increase user acceptance of INTACT over potentially competitive products because we plan to invest in understanding our users' performance, and the way they learn, in each domain of interest. To this end, Aptima has adopted a unique interdisciplinary approach to human-centered product development, integrating behavioral sciences, human factors engineering, software development, and marketing, from the initial requirement specification all the way to test and validation. Finally, it might be too early at this stage to have a pricing strategy for INTACT. We are not aware of any emergent pricing issue in the training/education market that might constitute an a priori barrier to market penetration. We plan to conduct such a market/pricing research study in the early stages of the Phase II project.