Technical Report 1179

Cooperative Interface Agents for Networked Command, Control, and Communications: Phase II Final Report

Scott D. Wood and Jack Zaientz
Soar Technology, Inc.

Carl W. Lickteig
U.S. Army Research Institute

April 2006

United States Army Research Institute for the Behavioral and Social Sciences

U.S. Army Research Institute
for the Behavioral and Social Sciences

A Directorate of the Department of the Army
Deputy Chief of Staff, G1

Authorized and approved for distribution:

MICHELLE SAMS
Technical Director

ZITA M. SIMUTIS
Director

Research accomplished under contract
for the Department of the Army

Soar Technology, Inc.

Technical review by

May Throne, U.S. Army Research Institute
Paula Durlach, U.S. Army Research Institute

NOTICES

DISTRIBUTION: Primary distribution of this Technical Report has been made by ARI. Please address correspondence concerning distribution of reports to: U.S. Army Research Institute for the Behavioral and Social Sciences, Attn: DAPE-ARI-MS, 2511 Jefferson Davis Highway, Arlington, Virginia 22202-3926

FINAL DISPOSITION: This Technical Report may be destroyed when it is no longer needed. Please do not return it to the U.S. Army Research Institute for the Behavioral and Social Sciences.

NOTE: The findings in this Technical Report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

REPORT DOCUMENTATION PAGE

1. REPORT DATE (31-01-06) April 2006	2. REPORT TYPE Final	3. DATES COVERED (from. . . to) May 2004 – October 2005

4. TITLE AND SUBTITLE Cooperative Interface Agents for Networked Command, Control, and Communications: Phase II Final Report	5a. CONTRACT OR GRANT NUMBER DASW01-02-C-0019
	5b. PROGRAM ELEMENT NUMBER 622785
6. AUTHOR(S) Scott D. Wood, Jack Zaientz (Soar Technology, Inc.), and Carl W. Lickteig (U.S. Army Research Institute)	5c. PROJECT NUMBER 211
	5d. TASK NUMBER A790
	5e. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Soar Technology, Inc 3600 Green Court, Suite 600 Ann Arbor, MI 48334 U.S. Army Research Institute for the Behavioral and Social Sciences 121 Morande Street ATTN: DAPE-ARI-IK Fort Knox, KY 40121-4141	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) U.S. Army Research Institute for the Behavioral and Social Sciences 2511 Jefferson Davis Highway ATTN: DAPE-ARI-IK Arlington, VA 22202-3926	10. MONITOR ACRONYM ARI
	11. MONITOR REPORT NUMBER Technical Report 1179

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

Contracting Officer's Representative and Subject Matter POC: Carl W. Lickteig

14. ABSTRACT *(Maximum 200 words)*: Report developed under a Small Business Innovation Research Program 2000.2 contract for topic A02-024. This Phase II research advanced the Phase I approach to enable improved human-system interaction of mixed human and robotic elements for a company-sized unit. The research reported here explored the utility of intelligent user interfaces for command and control tasks. A system prototype was developed using a virtual simulation environment, Soar-based intelligent agents, and a standards-based communications infrastructure. The prototype was evaluated by active duty Army officers using think aloud and situational awareness protocols conducted during a simulated urban mission. Results from the evaluation indicate that cooperative interface agents may be a practical technique for reducing command and control complexity, especially when manned and unmanned forces are integrated. Although this technique was demonstrated in a relatively simple simulation environment, further research is warranted to assess scalability and usability when applied to more knowledge-rich, real-world environments.

15. SUBJECT TERMS

Training, Interface Agents, Intelligent Agents, Command and Control, Future Combat Systems

SECURITY CLASSIFICATION OF			19. LIMITATION OF ABSTRACT	20. NUMBER OF PAGE	21. RESPONSIBLE PERSON (Name and Telephone Number)
16. REPORT Unclassified	17. ABSTRACT Unclassified	18. THIS PAGE Unclassified	Unlimited	112	Ellen Kinzer Technical Publications Specialist 703-602-8047

Technical Report 1179

Cooperative Interface Agents for Networked Command, Control, and Communications: Phase II Final Report

Scott D. Wood and Jack Zaientz
Soar Technology, Inc.

Carl W. Lickteig
U.S. Army Research Institute

Armored Forces Research Unit
Barbara A. Black, Chief

U.S. Army Research Institute for the Behavioral and Social Sciences
2511 Jefferson Davis Highway, Arlington, Virginia 22202-3926

April 2006

Army Project Number	**Personnel Performance and**
622785A790	**Training Technology**

ACKNOWLEDGEMENTS

This report describes work performed under a Small Business Innovation Research Program 2000.2 contract for topic A02-024. The project goal was to explore the use of intelligent agent technology to effectively control mixed human and robotic elements in the context of a future Future Combat System company. The success of this project reflects the efforts of many individuals.

First, we would like to thank the team at the U.S. Army Research Institute for Behavioral and Social Sciences (ARI) Fort Knox Field Unit. We especially thank Dr. James Lussier and Dr. Barbara Black for their support of both this project as well as our efforts to extend, apply, and commercialize this work in other ways.

We thank our subject matter expert, LTC (Retired) Scotty Abbott, U.S. Army. LTC Abbott's knowledge of Army future concepts, simulation, and armored operations made him an invaluable member of the team. During system evaluation he also played a critical role, assisting the evaluation team's interaction with active duty Soldiers and providing a key communication link between the research team and evaluation participants. Through his efforts we were able to glean many subtle details that might otherwise have gone unnoticed.

The support of active duty officers was also critical for the successful evaluation of our system prototype. Their insights, professionalism and dedication have well served those who will follow.

Finally we'd like to thank the entire Cooperative Interface Agent for Networked Command, Control and Communication (CIANC^3) project team for their exceptional efforts in making this project a success. We thank our development team, Jon Beard, Jacob Crossman, Dr. Rich Frederiksen, Laura Hamel, and Sean Lisse. Thanks also to our partner, Dr. Marcus Huber of Intelligent Reasoning Systems for his expertise in agent and multi-agent system theory. We thank our technical writing team, Frank Levy of Stories in Motion, Dr. Catherine Grasso and Kate Pierro. Thanks also to everyone else at Soar Technology who has contributed over the years.

COOPERATIVE INTERFACE AGENTS FOR NETWORKED COMMAND, CONTROL AND COMMUNICATIONS: PHASE II FINAL REPORT

EXECUTIVE SUMMARY

Research Requirement:

The purpose of this Phase II Small Business Innovative Research (SBIR) was to address the operational need to effectively command and control mixed teams of human and robotic elements. As robotic and automation technology improves, fundamental complexities in human-system interaction remain. It is clear that significant progress must also be made to improve the means by which human commanders interact with this new technology before its full benefit can be realized. There were three main goals for this project:

- Understand the requirements for human-system interaction at the company-command level in a realistic military scenario.
- Develop technology to enable improved human-system interaction of mixed human and robotic elements for a company-sized unit.
- Evaluate the developed technology with respect to effectiveness, usability, and training requirements.

Our basic approach to addressing the problem was to research, design, and develop intelligent user interface technology to assist battlefield commanders using the paradigm of intelligent software agents as a unifying concept. A graphical user interface was developed in a simulated environment using OneSAF (One Semi Automated Force) Testbed as the underlying simulation, and a formative evaluation was conducted with U.S. Army officers using a scenario derived from an FCS (Future Combat Systems) vignette as the overall evaluation task. While Phase I was demonstrated in a relatively simplistic context, demonstrating viability of the Phase II Technology under more realistic conditions required significant scientific progress in agent technology, agent-team collaboration, knowledge representation, and human-system interaction.

Procedure:

Under Phase I of this SBIR contract, the research demonstrated that a Soar-based intelligent agent approach was an effective means for implementing interface agents to facilitate sensor-shooter communications in a simple search-and-destroy scenario. The work-goal for Phase II was to further research the interface agent approach and to develop a viable prototype system that could serve as a test-bed for further human-system interaction research.

Under Phase II of this project, major accomplishments included:

- Scenario Definition and Requirements Analysis – A scenario using an FCS-company to assault an enemy compound was developed and the commander's tasks and necessary decisions were analyzed to determine sufficient interface functionality.

- System Architecture Design and Development – An architecture using the Control of Agent Based Systems (CoABS) grid agent environment was designed to facilitate agent communications and human-agent interaction.

- Agent Communication Development – A Foundation for Intelligent Physical Agent (FIPA)-compliant communications protocol was developed to enable structured, well-defined communications between agents, humans, and other system elements.

- Formal Agent-Interaction Protocol Definition – A formal deontic protocol was defined and implemented to simplify inherent agent design complexities, improve system robustness, and ensure verifiable agent system behavior.

- Ontology Research and Integration – An ontology is a formal knowledge representation of a particular domain that specifies objects, processes, relationships, concepts, and other entities. A mechanism was invented in this project to enable domain and doctrinal knowledge encoded within an ontology to be incorporated within Soar-based agents to simplify knowledge representation, maintenance, and consistency.

- Simulation Integration – The agent system and user interface were integrated with the OneSAF Testbed to enable rapid development and user execution of realistic scenarios.

- User Interface Design and Development – A commander's interface was developed for the scenario that included a combination of map-based display with a task-oriented mission display for organizing information.

- System Evaluation – A formative evaluation of the resulting system was conducted using U.S. Army officers running a simulated scenario. Metrics included successful mission completion and ability to maintain situation awareness. Post-evaluation questionnaires and interviews were used to obtain additional feedback.

Findings:

The system developed during Phase II allowed evaluation participants to successfully complete the evaluation tasks in a simulated scenario. The feedback received from participants was positive; generally the requests were for more of the types of automation provided by the $CIANC^3$ system. In some cases, participants wanted more control and less automation, but this was possibly due to evaluators intentionally limiting the complexity of the evaluation tasks. Another key observation was that the prototype system appeared to be a good platform for training and performing unmanned asset management: participants were able to effectively manipulate the position of multiple unmanned sensor assets in a way that maximized sensor coverage for the mission.

In many ways, the most important results are the Phase II advances in mixed-initiative technologies at the command, versus the operator, level. The key result in this area is that a triad of intelligent agents, Tasking, Coordinating, and Monitoring, can form the core of an intelligent user interface for command and control. These agents can work as a virtual command staff for users to reduce workload and simplify complex tasks. Another important result was the development of well-defined protocols for inter-agent communications and the establishment of responsibilities, permissions, and prohibitions for those agents. Finally, this project resulted in the development of bridge technology that connects ontologies with agent systems, a key enabler for future knowledge-rich intelligent systems.

Utilization and Dissemination of Findings:

The research conducted under this contract has been multi-faceted and generally applicable to many national security challenges. The project has demonstrated that intelligent user interfaces for emerging battlefield commanders is both possible and feasible. Technically, it has demonstrated that there are many benefits to implementing such systems using knowledge-rich, intelligent interface-agents. As with much research in human-system interaction, this work covers multiple disciplines, predominantly computer science and psychology. Thus, this report contains sections that may be of limited interest to purists in either discipline. The sections entitled “Introduction,” “Phase II Technical Objectives and Approach,” and “Conclusions and Phase III Transition Efforts” are of general interest and address the project as a whole. The sections entitled “Technical Background,” and “Phase II System Design and Implementation” will primarily be of interest to computer scientists and engineers. The section on “Phase II Usability Evaluation” will primarily be of interest to psychologists and human factors specialists.

The research and technology development conducted under this project have been successfully transitioned to other related research areas within the Army and Department of Defense. The Intelligent Control Framework (ICF) project is developing technologies to support context-sensitive control of robotic forces at the level of a robot operator for TARDEC (The U.S. Army Tank Automotive Research, Development, and Engineering Center). The Robotic Command and Control Intelligent Enablers (ROCCIE) project, under CERDEC (The U.S. Army Communications-Electronics Research, Development, and Engineering Center), is researching techniques for combining different types of reasoning and knowledge systems like planners, intelligent agents, and ontologies to make intelligent support systems more capable and better able to interoperate. The Knowledge Enablers for Unit of Action (KEUA) project, supported by the Army Research Laboratory (ARL) sought to develop agent technology that would help facilitate the understanding of battlefield information to enable better decision-making. The Battlespace Information and Notification through Adaptive Heuristics (BINAH) project, supported by the Office of Secretary of Defense and the Air Force Research Laboratory is developing intelligent support for information delivery and visualization for intelligence analysts. The High-Level Symbolic Representation (HLSR) project under the Office of Naval Research (ONR) is developing engineering techniques for simplifying and improving languages for creating intelligent systems.

The research and development conducted in this project has demonstrated that intelligent support systems can be an important technique for reducing system complexity for the warfighter, while improving human performance and mission effectiveness. This work uncovered many issues that warrant further investigation and developed general techniques that are applicable to a wide variety of commercial and defense challenges.

COOPERATIVE INTERFACE AGENTS FOR NETWORKED COMMAND, CONTROL AND COMMUNICATIONS: PHASE II FINAL REPORT

CONTENTS

Page

Page

List of Tables

List of Figures

COOPERATIVE INTERFACE AGENTS FOR NETWORKED COMMAND, CONTROL AND COMMUNICATIONS: PHASE II FINAL REPORT

Introduction

In Joint Vision 2020 (JV2020), the Department of Defense describes the operational concepts necessary to face the wide range of interests, opportunities, and challenges that will be required of the United States military to both win wars and contribute to peace. As part of this vision, there is a massive transformation underway that trades steel for information, calls for large numbers of unmanned sensors and vehicles, and depends on a rapid tempo of operation and a mutual understanding of the global situation at all echelons.

Concepts including joint command and control, precision engagement, and information operations, represent additional complexity for warfighters and will require significant technical breakthroughs to realize their full potential. Specifically, JV2020 describes the need for improved battle command capabilities, noting that faster operational tempo, increased choices among weapons, and greater weapons ranges will require continuous, simultaneous planning and execution at all levels. In response to this need, JV2020 calls for the development of new, highly automated supporting tools for commanders to enable flexible, adaptive coordination of both manned and unmanned systems.

To address these needs in a way that improves warfighter performance, rather than adding to warfighter workload, requires the development of significantly smarter control and information systems. Such systems should, at minimum, accept delegated tasks, monitor significant events, and speed the transformation of data into understanding.

While there are many possible approaches to developing smarter systems, the Cooperative Interface Agents for Networked Command, Control, and Communications ($CIANC^3$) project focused on the creation of intelligent human-system interfaces designed to simplify and augment warfighter interaction that can function as a layer on top of existing as well as future battle command and information systems. Figure 1 shows the $CIANC^3$ system prototype developed under Phase II of this project. The $CIANC^3$ system incorporates intelligent agent software to implement an intelligent user interface for command and control of mixed human and robotic units. The system was evaluated at Fort Knox using active-duty officers from the U.S. Army.

This report discusses the need for intelligent assistance and decision aids, research and implementation of the $CIANC^3$ system, and the formative evaluation. The report concludes with a discussion of implications for future research regarding intelligent user interface design, development of intelligent multi-agent systems, training for future command and control, and operational issues regarding the deployment of intelligent military systems.

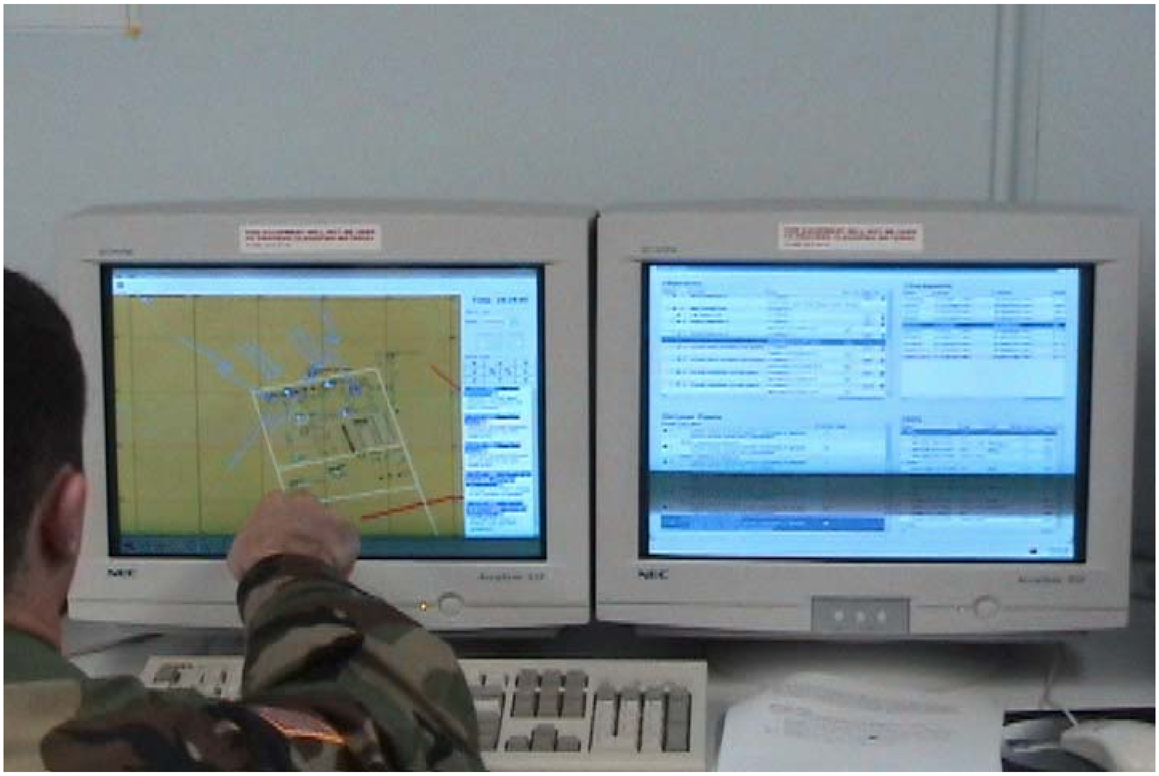


Figure 1. An evaluation participant using the CIANC[3] interface.

Identification and Significance of the Problem

There are many challenges to creating intelligent human-system interfaces, including understanding the operational needs and specific human limitations which intelligent interfaces can augment, conducting the basic research and developing the technological infrastructure necessary to create a prototype, and integrating interface components with command and control systems (or prototypes) to understand which aspects contribute most to improved warfighter performance, and why. While each of these challenges is significant in its own right, the approach here has been to explore a very narrow vertical slice through each, rather than exhaustively explore each level prior to addressing the next. This methodology has been instituted in order to demonstrate the viability of intelligent warfighter interfaces and, more generally, to build the foundation for a more comprehensive effort. An additional benefit of demonstrating how intelligent warfighter interfaces can be applied in practice is that it will enable others to envision new applications.

The focus of the CIANC[3] project has been on robotic command and control, for which this project's researchers have identified human-system interaction problems, designed potential solutions, and created intelligent agent software that supports the commander's tasks as well as mitigating human performance limitations. The U.S. Army's vision for Future Combat Systems (FCS) includes the use of mixed teams of human and robotic forces on a dynamic battlefield. Implementing this vision will require a shift from manual control of weapons, to semi- and fully automated control of entire teams of human and non-human entities. It will also entail an overall force reduction, with multiple entities controlled by individual team leaders and multiple teams led by higher-echelon commanders.

To accomplish this, systems will have to be designed to require less human interaction and greater robotic autonomy. Successful implementations will incorporate autonomous and semi-autonomous robotic forces in a command and control infrastructure that allows human, robotic, and mixed teams alike to be controlled quickly and easily. One key to success is the degree to which teams and individual robots are autonomous. A second key is whether the commander's human-machine interface is designed so that the commander is not overloaded with constant system interaction and can focus on his or her mission.

Phase II implemented an agent architecture based on decomposing the command and control problem into three main task areas: Monitoring, Coordinating and Tasking. By using agents that specialize in each of these three areas as an interface to the underlying robotic behaviors, researchers were able to develop an intelligent interface that can assist company-level commanders to command multiple teams of human and robotic elements. One key objective in this work has been to develop software techniques and technologies that allow commanders to control the robot teams the way they command human teams — that is, in the language of the military, not the language of robotic control theory.

Warfighter Need for Intelligent Interfaces

In observations of warfighter interaction and other research using prototype battle command systems (e.g., Lickteig, Sanders, Lussier, & Sauer, 2003), this project identified several key areas where some form of intelligent automation might be useful. These can be divided into two categories: understanding the environment and manipulating the environment. Understanding the environment means having sufficient awareness of the current situation to enable sound decision-making and effective actions. For new information this means recognizing when new information is significant, how it fits with currently available information, and how that information will change the current situation (i.e., Level-3 Situation Awareness; Endsley, 1988). The process of actively understanding the environment can be formally characterized as Battlefield Visualization (FM 6-0). Battlefield Visualization is a three-step command process whereby the commander develops a clear understanding of the current situation, envisions a desired end state, and visualizes the sequences of activity that will move his force from its current situation to the desired end state. While understanding the environment is critical for effective command, the main focus of this work is on manipulating the environment.

Manipulating the environment can be viewed as giving commands to subordinate elements, coordinating and synchronizing the operation of multiple elements, and adjusting existing plans as necessary during the execution of an operation. In human-to-human operation, such as from a commander to his or her staff, or from a commander to subordinate units, often only intent is conveyed (or even necessary). From that intent the recipient adds available context (or requests additional information) that is used to develop an actionable plan. While performing this transformation from intent to action can be very direct among experienced warfighters, automating it to occur without human assistance can be very difficult. As such, human-robot interaction or human interaction with other automated systems can only be done at a very basic level, where each detail must be clearly specified. It is these "common sense" inferences that make human-robot interaction so workload intensive, especially when re-planning (or plan adjustment) is nearly constant. Coordinating the actions of multiple unmanned elements, say between a sensor and shooter, further compounds warfighter effort. Performing multiple such tasks, especially when stretched and interleaved over time, more dramatically increases the cognitive demands on the warfighter and increases the probability of catastrophic errors. Simplifying the transformation of command intent and facilitating the coordination of multiple unmanned elements is a primary operational focus of efforts to address the warfighter's need to manipulate the environment.

In current operational environments, human experts are used to solve the challenges described, by bringing to bear years of experience and knowledge. Developing automated solutions that approach or exceed human capabilities, and that can do so in a dynamic, hostile environment, will require an equally large set of expert knowledge. This knowledge includes the patterns of information that experts use to identify problems and solutions, the analytical processes and heuristics that experts use to approach and solve problems, and the reasoning that experts use to evaluate information when that information is uncertain or incomplete. The approach in this research is to encode expert knowledge into agent-based systems that can be combined dynamically to form intelligent user interfaces, and applied to a wide variety of circumstances and purposes. Key to developing such intelligent solutions that augment rather than hinder human performance is developing a deep understanding of how humans interact with automated and intelligent systems.

Summary of Phase I

The purpose of Phase I of this project was to demonstrate the feasibility of using a multi-agent framework to facilitate human-robot interaction within a sensor-shooter scenario. The approach was to develop a simplified version of the agent infrastructure and integrate it with a modified version of OneSAF Testbed Baseline (OTB 1.0) used for robotic control known as the Operator Control Unit.

The agent and communication system designs were successfully implemented in a simulation environment. A scenario was created to test the system using a simple combination of a sensor-vehicle Unmanned Aerial Vehicle (UAV), and a shooter-vehicle Unmanned Ground Vehicle (UGV). The UGV was tasked to seek and destroy a suspected enemy. The UGV tasked a UAV to locate and acquire the target. The UAV located the target and transmitted the coordinates to the UGV, which then confirmed with the human operator before firing on and destroying the target.

The scenario was simple enough to test and demonstrated the capabilities of the interface-agent architecture, but it was not complex enough to demonstrate any real utility to robotic control. In addition, the Tasking agent accomplished most of the background work. A more complex scenario would place more demands on the Coordinating and Monitoring agents, driving their further elaboration. An additional finding was that inter-agent communication patterns could quickly become complex and unwieldy, even for simple scenarios. Unified Modeling Language (UML) sequence diagrams were used to help simplify the communications design.

The performance of the Soar cognitive architecture was more than adequate for the agent task in the simple scenario. However, implementing the agent behaviors was somewhat complex, even with that architecture. Phase I resulted in the identification of multiple technology gaps:

1. More research effort is needed to develop tools and techniques for rapid agent development.

2. The large amount of declarative knowledge that needs to be encoded into knowledge intensive systems can be overwhelming. Representing such knowledge within production rules inhibits scaling, because changing the knowledge is time-consuming, expensive, and error-prone work. Ontologies and similar forms of knowledge representation are needed to disentangle procedural (behavioral) knowledge from declarative knowledge, making such systems easier to develop and maintain.

3. Although FIPA provided a great foundation for developing the agent communication infrastructure, alone it cannot meet the inter-system and inter-agent communication needs of military systems. Phase II should explore grid-based computing and communication content languages.

In summary, Phase I successfully demonstrated the technical feasibility of interface agents for robotic command and control. It also provided infrastructure and techniques necessary to rapidly explore much more of the problem space in Phase II.

Phase II Technical Objectives and Approach

Phase II effort focused on developing the fundamental architecture to demonstrate the viability of an agent-based approach to supervisory command and control, and to facilitate continuing research. To do this, the research team developed a very narrow set of functionality for a limited operational scenario. As planned, this approach resulted in a modest demonstration of new capabilities, yet has made apparent many of the challenges to implementing network-centric solutions (independent of whether the approach is agent-based).

The technical objectives for this SBIR were to demonstrate the feasibility of a $CIANC^3$-like system for control of battlefield robots. That is, the project aimed to determine whether an agent framework built around the three specified agent types could be constructed to add an intelligent abstraction layer between human military commanders and robotic battlefield entities. Phase I demonstrated feasibility on a technical level. Phase II tested whether such a system might actually benefit FCS commanders. The technical objectives were:

- Determine human information needs for controlling mixed human and robotic teams.

- Determine appropriate levels of automation for human tasks that will reduce cognitive workload yet maintain sufficient human control.

- Develop a suitable high-level architecture for agent organization and develop an inter-agent interaction protocol.

- Develop a usable human interface to software agents that will demonstrate agent interactions, demonstrate abstract-to-concrete command translation, and allow testing of target scenario.

- Determine scalability of a prototype system and develop a more complex scenario to demonstrate these capabilities.

- Further demonstrate the feasibility of the cooperative agent concept and explore real-world issues by integrating the prototype technology into a commander's interface, linked to a virtual simulation system.

- Evaluate the system for usability and performance, using a variety of engineering and psychological techniques.

The approach taken to achieve the technical objectives was to create a framework of cooperative interface agents for networked command, control, and communication. This was begun by augmenting the current roles found in command staffs. These roles were then extended to provide real-time situation awareness (e.g., Endsley, 1988) and decision support beyond what is humanly possible. Command staffs commonly serve five basic functions to commanders in support of reconnaissance, security, offensive, and defensive operations:

- Provide timely and accurate information.

- Anticipate requirements and prepare estimates.

- Determine courses of action and make recommendations.

- Prepare plans and orders.

- Supervise execution of decisions.

To assist in the automation of routine tasks, small, encapsulated software agents can often be used to perform much of the tedious parts of user tasks. Such agents are often referred to as intelligent agents, or rational agents (Wooldridge, 2000), and have been used to assist users with tasks such as scheduling meetings and purchasing products, and for other intelligent user interfaces. While some agents operate solely in the background, interface agents are designed as user interface elements that can directly assist users with their tasks. This can include assisting with the specification of complex commands during input tasks to decrease task execution time and improve accuracy. Interface agents can also assist with information output, interpreting raw data or filtering necessary information from non-relevant data.

A weakness of some of the previous work on intelligent interface agents is that human operators needed a significant amount of training to use them and they had to think in terms dictated by the software agents. A goal of intelligent interface design is to make the interface invisible (Maes, 1994). This can best be accomplished by merging software agent technology with proven direct manipulation techniques such as window scrolling and other desktop metaphors embodied in modern graphical user interfaces.

To demonstrate the feasibility of using intelligent interface agents, this project tested if it could provide the functionality currently provided by command staffs. These functions were divided among three classes of agents: tasking, monitoring, and coordinating. Figure 2 illustrates a notional organization for how the interface agents might work in the larger scheme of battlefield command and control. Here, a cluster of intelligent software agents acts as an intermediary between a system user and some collection of complex technology. In the illustration, a warfighter within a command vehicle uses an intelligent system that provides context-driven display and task assistance using a team of cooperative interface agents embedded within the system. In addition, it is assumed that these interface agents would have access to, and be integrated tightly with, other battlefield information and decision support systems. Although other solutions are possible, the need for rapid tasking, coordinating, and monitoring of operations will remain, irrespective of the type of digitized services that will become available to battlefield commanders.

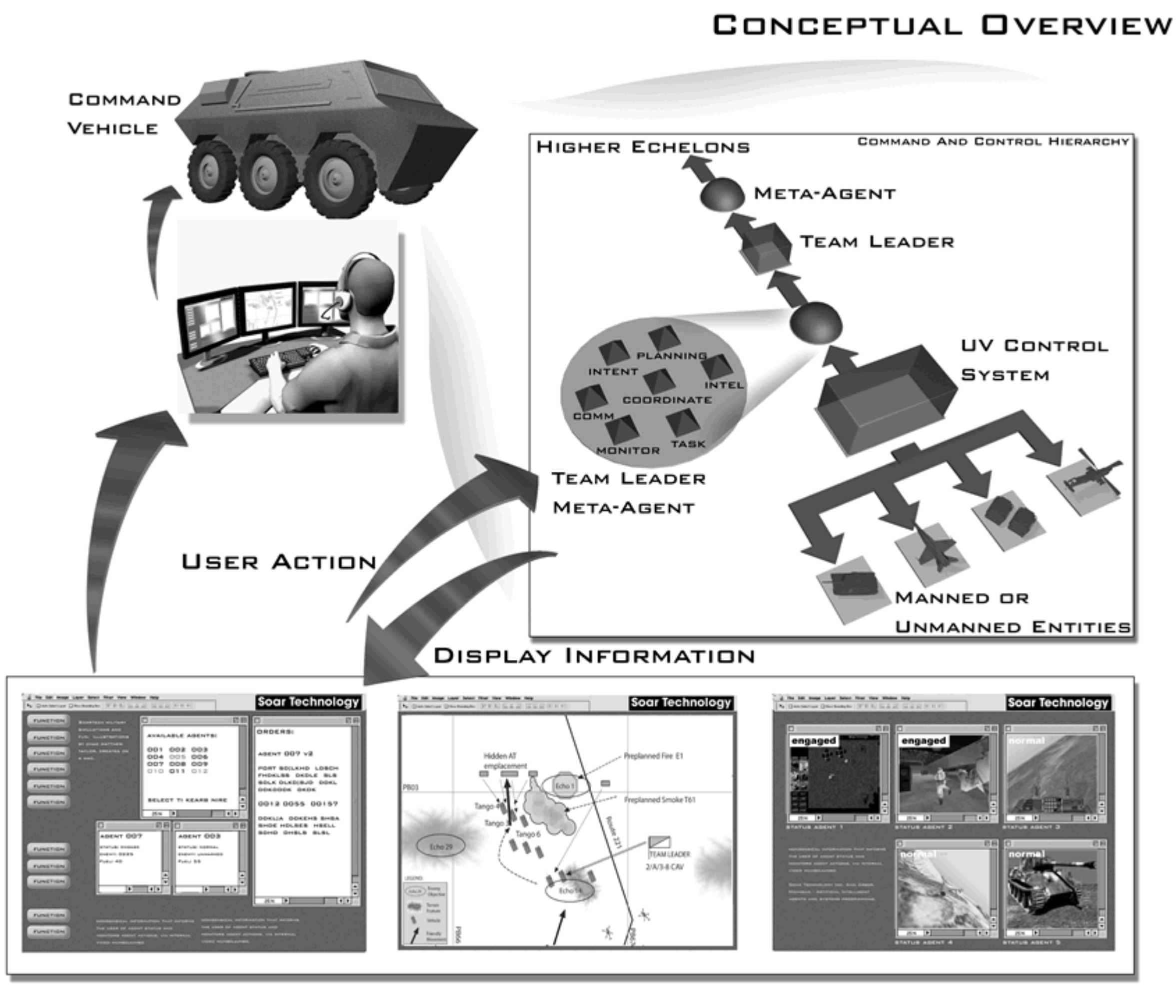


Figure 2. CIANC^3 conceptual overview within networked environment.

Conceptual Scenario

The CIANC[3] project was conducted in support of the U.S. Army's FCS program, exploring agent-based technologies for systems that do not yet exist and doctrine that has not been fully developed. The goal was to develop a scenario that could not be completed without some form of automated assistance. The conceptual scenario was based on the FCS Unit of Action Baku vignette, using a company-level blue force equipped in a way similar to that of an FCS Reconnaissance company (Note: these vignettes are constructs designed to act as snapshots of the FCS employed in combat operations). Figure 3 shows an artist rendering of the conceptual scenario. In this scenario, a single operator is coordinating an assault on an enemy compound using a mix of unmanned ground and air vehicles, as well as conventional troops.

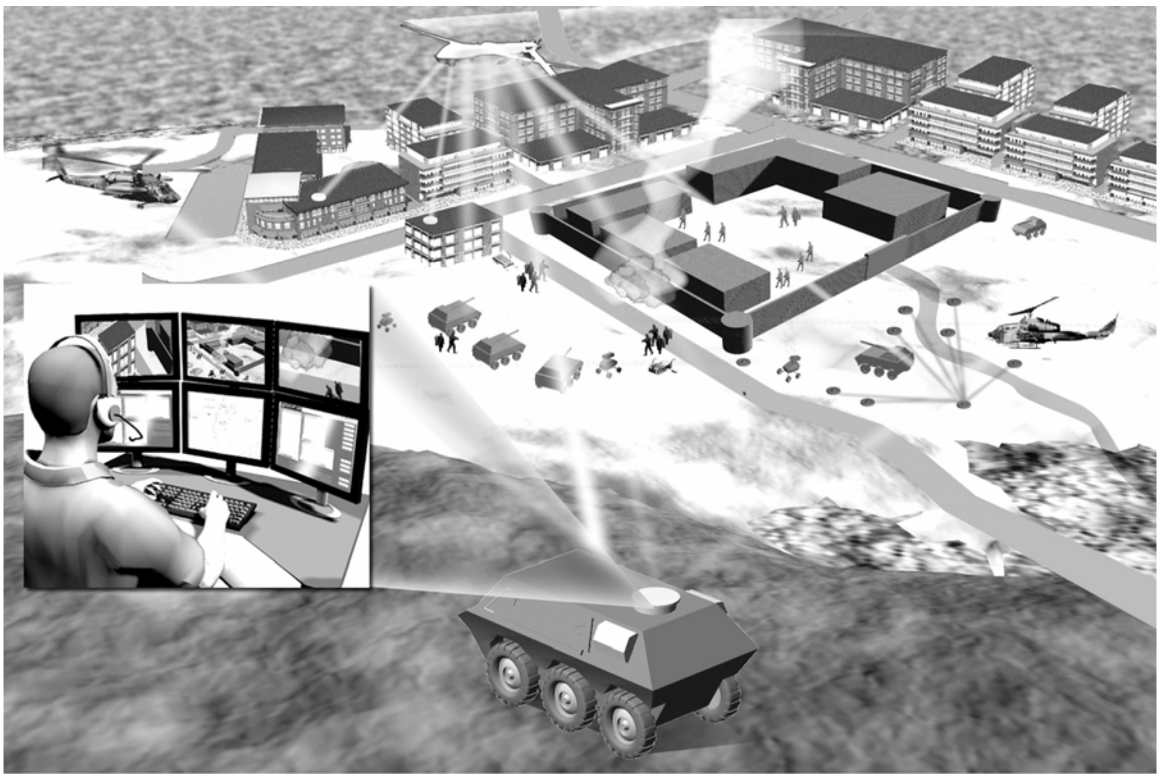


Figure 3. CIANC[3] conceptual scenario integrating human and robotic forces for urban assault.

The FCS company is tasked to breach a walled urban compound and secure the area. A mixed human-robot FCS company assaults a red force. The scenario is currently implemented using the Joint Semi-automated Force (JSAF) simulation environment. The assault follows four phases: condition setting, movement to a position of advantage, seizure of objective, and secure until relieved. Specifically, the plan calls for an initial placement of Unmanned Air Vehicles (UAVs) in key reconnaissance positions, movement of ground assets into breach position, wall breach, and ground-based assault.

Technical Background

This section provides necessary background on robotic entities, human-system interaction, intelligent user interfaces, intelligent interface agents, multi-agent systems, and the operating environment for future agent-based systems.

Robotic Battlefield Entities

An overall goal of the FCS program is to transform the current military structure, operations, strategies and tactics to create a force that is more responsive, deployable, agile, versatile, lethal, survivable, and sustainable. One strategy for achieving this goal is to split the roles of battlefield entities to create smaller, more specialized platforms that will operate cooperatively in a much more effective manner than currently possible. This will include at least the following battlefield platforms: manned vehicles, direct fire vehicles, indirect fire, beyond line of sight (BLOS) vehicles, sensor vehicles, unmanned aerial vehicles, and other layered sensors such as satellites (c.f., U.S. Army, 2005). Other research is addressing low-level issues regarding autonomous robot control, such as cooperative path planning, team selection and tactics, and dealing with uncertainty (e.g., Defense Advanced Research Project Agency's (DARPA) Coordinators program and the Army's Future Force Battle Command Integration initiative). The present work developed software techniques and technologies to allow human commanders to control the robot teams similar to how they command human teams – that is, in the language of the military, not the language of robotic control theory. It also addressed command and control for higher echelons and for cooperative actions across echelons.

Human-Machine Interaction and Intelligent User Interfaces

An overall goal of the human-machine interface design for this project was to maximize human performance by creating a system that allowed users to focus on the military objectives rather than on the technological means for accomplishing those objectives. This required a system that is highly usable: efficient to use, easy to learn, easy to remember, error-tolerant, and subjectively pleasing (Brinck, Gergle, & Wood, 2001). Two approaches that have been taken to improve usability are direct-manipulation interfaces and intelligent interfaces. Direct manipulation interfaces stress the ability of users to directly, and naturally, manipulate and navigate their environment using metaphors of the physical world such as desktops, folders, and trash cans. This approach has been successfully applied to the visualization of large datasets and is the basis for most modern graphical user interfaces.

Another technique for improving usability is by developing intelligent user interfaces to automate mundane and time-consuming tasks. Previous efforts at automating system tasks have achieved mixed results, often because supervisory control issues were not adequately addressed (Leveson, 1995; Sheridan, 2000). Effectively automating system functions requires achieving a delicate balance between reducing tedious tasks along with overall operator workload, and maintaining adequate human vigilance and control (both real and perceived). For example, users can become complacent in monitoring-only tasks, such as monitoring status gauges or security cameras, and become more prone to errors. They need to be kept engaged and to maintain their skills for times when automated systems are inadequate. Task-analytic techniques can be used to address the supervisory control problem, enabling designs that include the right mix of human and automated control (Wood, 1999; Wood & Kieras, 2002). One way of implementing supervisory control software is through an intelligent user interface.

The term intelligent user interface describes a broad class of system types that can apply artificial intelligence techniques to any aspect of human-system interaction. Historically, intelligent user interface meant an expert system. The approach was typically to encode a large amount of expert knowledge into one knowledge base, forming large decision trees of if-then rules. The users, often experts themselves (such as doctors), either would engage in a dialog where the system asked a series of questions, or would prepare the set of available data so that it could be entered into the system. The intended result was for the system to diagnose a problem or answer questions that their less-informed users could not. This class of system was thus dubbed the "Greek Oracle" approach (Miller & Masarie, 1990). Such expert systems suffered from three key flaws. First their knowledge base was fragile, meaning that they didn't deal well with information they were not specifically programmed to provide. Second, users found them difficult to use, especially in time-critical situations (such as medical diagnosis). Third, and perhaps most important, expert systems were not designed to capitalize on human strengths. Instead they sought to replace the creativity and pattern-matching skills that are key human strengths. They relegated the users to the menial task of feeding info to the system. Hence, even though some very capable expert systems were created, they failed to gain general acceptance because they did not represent a suitable paradigm for human use.

More recently, much effort has gone into understanding how intelligent systems can be used to support the user's task while fitting into the user's domain, rather than the other way around. Roth, Malin, & Schreckenghost (1997) characterize these efforts as representing three broad paradigms:

- Intelligent Interfaces as Cognitive Tools – Cognitive tools are designed to augment the mental abilities of users, not by providing all the answers, but by helping to formulate the questions, gathering necessary information, and managing complexity to avoid data overload. Examples include aerospace fault management systems (Malin et al., 1991) and next-generation medical reference systems (Miller, 1986).

- Intelligent Interfaces as Elements of Cooperative Systems – Cooperative system elements include agent-based systems, such as interface agents (Maes, 1998), that function as part of a human-agent team for accomplishing cognitive tasks (Hutchins, 1995). Such elements serve a critical role in creating mixed-initiative interaction interfaces where control and responsibilities shift dynamically between human and agent (cf., Horvitz, 1999).

- Intelligent Interfaces as Representational Aids – Representational aids focus explicitly on the problem of displaying information, often from different sources and in different mediums, to the user in a way that facilitates rapid understanding and sense-making. Such aids can dynamically configure information delivery according to user task, user state, concurrent events or other contextual information specific to the user's situation.

These categories roughly correspond to the traditional human-computer interaction notion of model-view-controller (MVC) where representational aids assist with viewing and perceiving relevant aspects of the model, cooperative elements assist with controlling the system and manipulating the model, and cognitive tools assist with understanding the model. Following the MVC analogy, it would not make sense for an operational system to contain only a subset of the three paradigms. Just as it would not make sense for a traditional software application to contain only a model (e.g., database) and controller (e.g., keyboard), but no view (e.g., display window), an operational intelligent user interface would likely contain aspects of all of these paradigms (e.g., Maybury & Wahlster, 1998). One way of implementing intelligent user interfaces is through intelligent interface agents.

Interface Agents

Interface agents (Laurel, 1990) are a specific form of software designed to reduce the complexity of human-system interaction. Such agents can take the form of relatively simple agents for performing single, well-defined tasks such as filtering mail, or they can be fairly complex for more complicated tasks such as seeking out useful information or web sites (Lieberman, 1997). Fundamentally, interface agents represent an additional, simplifying layer of abstraction between a user and a computer system.

Agents provide the interface with the capacity for a mixed-initiative dialog allowing for the more natural give and take characteristic of typical human conversation. Key elements of this dialog (Horvitz, 1999) include the interface agent's ability to:

- Consider uncertainty about the commander's goals.
- Consider the status of the commander's attention in the timing of services.
- Infer ideal action in light of costs, benefits and uncertainties.
- Employ dialog to resolve uncertainties.
- Allow direct invocation and termination of interface services.

This dialog between commander and system will provide a flexible level of control that can adapt to the dynamic environment of battlefield command, offering the commander as little or as much direct involvement as is required by situation, doctrine, or commander preference.

Benefits of the Agent Paradigm

Using an agent paradigm allows researchers to approach computer-based, complex-problem solving in a way similar to how one would employ human teams to solve complex problems. Instead of developing or utilizing a single problem-solver (human or computer) to reason about large, complex challenges, teams of experts can be formed (human or agent) to dissect the problem and solve it cooperatively. This approach not only allows researchers to utilize a broader range of deeper knowledge, but also permits reuse of that knowledge by enabling different team configurations that are problem-specific. As with humans, creating an effective team also requires developing effective communication protocols and rules of interaction. This approach is being widely researched throughout Department of Defense (DoD) organizations as an alternative to traditional, inflexible software engineering.

It is important that interface agent technology be developed modularly, creating cohesive, loosely coupled entities that can be easily adapted as doctrine, technology, and missions evolve. Dividing agent workload between a set of specialized modular agent types provides a number of key benefits.

Encapsulation of Knowledge

Localizing doctrinal knowledge (e.g., tactics, techniques and procedures) in specialized agents provides a natural mechanism for matching interface-processing rules with military doctrine. Agents that will be part of the DoD's Command, Control, and Communications (C^3) structure must adapt to changes in doctrine over time as well as by service and operation. As requirements change, agents encapsulating the new rules can be introduced into the system without impacting other aspects of the system.

Encapsulation of Processing

Localizing task execution in specialized agents also provides a natural mechanism for encapsulating processing and distributing computation. As the duties of the individual $CIANC^3$ agents increase in scope and sophistication, specialized techniques will be adopted or developed to increase task performance, robustness, or scalability. While current research utilized the Soar architecture for agent decision-making, it is likely that future $CIANC^3$ agents will require the addition of dedicated planners, case-based reasoning systems, and other AI technology.

Communication-Oriented Design

It is important to note that the division of knowledge and processing into distinct agent types creates a demand for a more sophisticated communication infrastructure than might be required by a monolithic system. This increased sophistication, despite the additional development requirements to construct it, is another one of the key benefits of the system because it supports a more natural, modular architecture. Establishing this capacity as a fundamental characteristic of the architecture allows the seamless introduction of new processing or reasoning components at any time or at any location in the $CIANC^3$ architecture.

Reconfigurable Design

It should also be assumed that the target agent organization described here will change to include other classes of interface agents. The agent architecture, therefore, must accommodate such change. For example, a display agent could be used to control all information presented to the user. An executive agent may be useful for coordinating the control and communication within a collection of agents (e.g., within a meta-agent). Other agent roles that might be separately developed include:

- Deriving the commander's current task from recent actions.

- Deriving enemy intent based on recent enemy actions.

- Evaluating and critiquing plans.

- Routine scheduling of communications, supply, and duty rotations.

Additionally, the missions, roles, responsibilities and information requirements will be different for each echelon in which this technology is employed. Doctrine will also change with coming technological advances. It is important that the resulting system be flexible and modular enough to rapidly adapt to new procedures and protocols. For example, the agent system should be constructed to allow different sets of expert knowledge to be easily constructed and integrated into the agents.

Multi-Agent Systems

There are many challenging issues that must be addressed when developing multi-agent systems. This includes how the agents are organized and what role the agents play within the organization (Birmingham, D'Ambrosio, Darr, & Durfee, 1994; Fox, 1988). Within the DoD systems, much of the agents' organization will be dictated by military doctrine. However, with multiple agents associated with each unmanned vehicle (UV) operator and the possibility of combat losses, it will be important to address static and dynamic organization and role determination (Corkill, 1982; So, & Durfee, 1994; So, & Durfee, 1997).

Another important issue in multi-agent systems is determining what communication language semantics and syntax the agents will use at both the *performative* and *content* levels (FIPA, 2000; Labrou, 1996; Cohen, & Levesque, 1990c; Huber, 1999). The performative level is associated with the intention of the message, such as whether it is a directive (command, question, or request), an assertive (information/knowledge passing), a commissive (commitment forming), etc. (Searle, 1970). The content level is associated with the specifics of the communication, such as the task being requested or the information being passed, and is almost always domain specific.

Entities within organizations tend to interact with each other in standard patterns, and this holds true for intelligent agents as well. These interaction patterns simplify agent reasoning by constraining agent behavior, and facilitate the creation of expectations and standard behavior models in other agents. Capturing these patterns, commonly called conversation policies or interaction protocols (Bradshaw, Dutfield, Benoit, & Woolley, 1997; FIPA, 2000; Kumar, Cohen, & McGee, 2001; Labrou, & Finn, 1997), is required in any complex multi-agent environment and needs to reflect, for example, any authority relationships that exist between agents (Jones, & Sergot, 1996).

The manner in which the agents work together to complete their tasks is crucial to the agents' performance in any domain, and has been the topic of a great deal of research. There are many factors involved with determining the problem-solving paradigm of the multi-agent system. Just a few issues include whether problem solving is done in a centralized or decentralized manner (Fox, 1988; Durfee, Kenny, & Kluge 1998), whether tasks are distributed or can be handled by a single agent (Gasser, & Hill, 1990), the level of robustness and fault tolerance required in the domain (Kumar, & Cohen, 2000; Rosenschein, 1985), the level of uncertainty and rate of change in the environment (Fox, 1979), whether a static problem solving scheme will be used or whether the problem solving scheme can be dynamically changed (Decker, & Lesser, 1995; Rosenschein, 1985).

Operational Environment: Service-Based Architecture Requirement

Information Age transformation requires an understanding of how various technologies can fundamentally improve mission effectiveness. This requires not only an understanding of the technology, but an understanding of how Soldiers can best use that technology, and how that usage can fit into or transform military doctrine. At the core of military doctrine is the Military Decision-Making Process (MDMP); a methodical, deliberate analytic process for problem solving that pervades all military operations. If transformation is to truly represent a revolution in military affairs, it must enable fundamental improvements to the MDMP and tactical decision-making. One evolutionary change to MDMP is the move towards a running estimate of battlespace information that will allow more rapid assessment, awareness, and understanding of the situation. The goal of this change is to ensure information superiority, enabling more rapid decision-making and resulting in more decisive battles. For example, the development of the Global Information Grid (GIG) will vastly increase the amount of information available to all echelons of command and will allow information sharing and collaboration to be conducted in a peer-to-peer manner. This will enable information to break beyond the bounds of the traditional command hierarchy, in effect, pushing the power of information to the edge of the force network. To the warfighter, this means the empowerment that more information provides, but also the burden of making sense of that information. Developing the technology that will allow warfighters to rapidly understand and process large amounts of rapidly changing data are critical to realizing the Network Centric Warfare (NCW) vision of dramatically increased mission effectiveness, self-synchronization, improved information sharing and collaboration, and an improved, shared situation awareness (Alberts, Garstka, Hayes, & Signofi, 2001).

Phase II System Design and Implementation

The prototype developed to support the FCS Unit of Action Baku scenario was designed to provide entity-level control and coordination based on a commander's Operational Orders (see Figure 4). The goals of the demonstration prototype were:

- Show reasoning over simulated entity capabilities and disposition, rules of engagement, the current operating scenario, and commander's intent.

- Task and coordinate networked sensors, maneuver, and effects in real time.

In this scenario, the FCS company is tasked to breach a walled urban compound and secure the area. The assault follows four phases: condition setting, movement to a position of advantage, seizure of objective, and secure until relieved. Specifically, the plan calls for an initial placement of UAV's in key reconnaissance positions, movement of ground assets into breach position, wall breach and ground-based assault.

Figure 4. CIANC[3] urban assault scenario with FCS simulated company.

User Interface Analysis and Prototype

To address the objective of developing an effective user interface for robotic control, the research team first had to determine warfighter information needs. The target user for this system was a company commander or a subordinate who would be responsible for commanding and coordinating human and robotic forces, but not necessarily directly controlling them. The researchers started by first developing a detailed system usage scenario based on current doctrine and equipment. Then the new platform and weapons capabilities were projected onto the FCS scenario to determine how this might change or affect the target user's command task.

The prototype interface was then designed to support the resulting task. This involved two key assumptions:

- Irrespective of new technologies, fundamental tenets of command and control are unlikely to change dramatically.
- To keep from imposing an additional workload burden on the user, human-robot interaction should be at least as easy as human-human interaction.

Usage Scenario

The following usage scenario was developed to analyze how a warfighter might use the prototype system while conducting the scenario mission within the evaluation environment. The scenario is divided into distinct phases including staging, pre-operation, operational, and post-operation:

Staging Phase tasks begin with receipt of Operational Orders (OPORD) and mission briefing, and analysis of data from numerous sources including intelligence reports, maps, and other available information. From this data, information is developed, correlated and displayed on system displays. This will enable more accurate situation awareness to be developed and maintained regarding friendly, enemy, and civilian positions and courses of action. Pre-Operation Phase tasks then follow with analysis of mission goals, plan development, and plan approval. The Operational Phase commences using the system graphical user interface (GUI) to issue commands, communicate, receive reports, and make tactical decisions as necessary. The initial plan in this Operation places three UAVs at recon points with the expectation that there may be UAV losses. Each loss triggers a notification that is matched against a pre-set loss threshold. When this threshold is in danger of being crossed, the user is warned. The user can choose to change the ratio, move or delete recon points, or ignore. Operations continue with the user issuing orders to subordinate units via the GUI to conduct movement, breaching, and assault tasks to successfully accomplish the mission. The Post-Operation Phase includes debriefing and an after-action review.

From the usage scenario, eleven general-purpose GUI tasks were defined to enable a user to perform the necessary tasks using the prototype system. For each GUI task, assumptions were listed and corresponding user and system behavior was specified. Using the set of GUI tasks from this list, the user could execute all of the evaluation tasks.

GUI Screen Design

From the usage scenario and GUI task definitions, a two-screen user interface was designed. The first screen, the Map Display (Figure 5), was designed around a simulated view of the battlefield, in this case using the OTB simulation environment. It includes mission control widgets for starting and stopping the simulation, map navigation controls, and a scrolling message window where the system and simulated entities can communicate with the user.

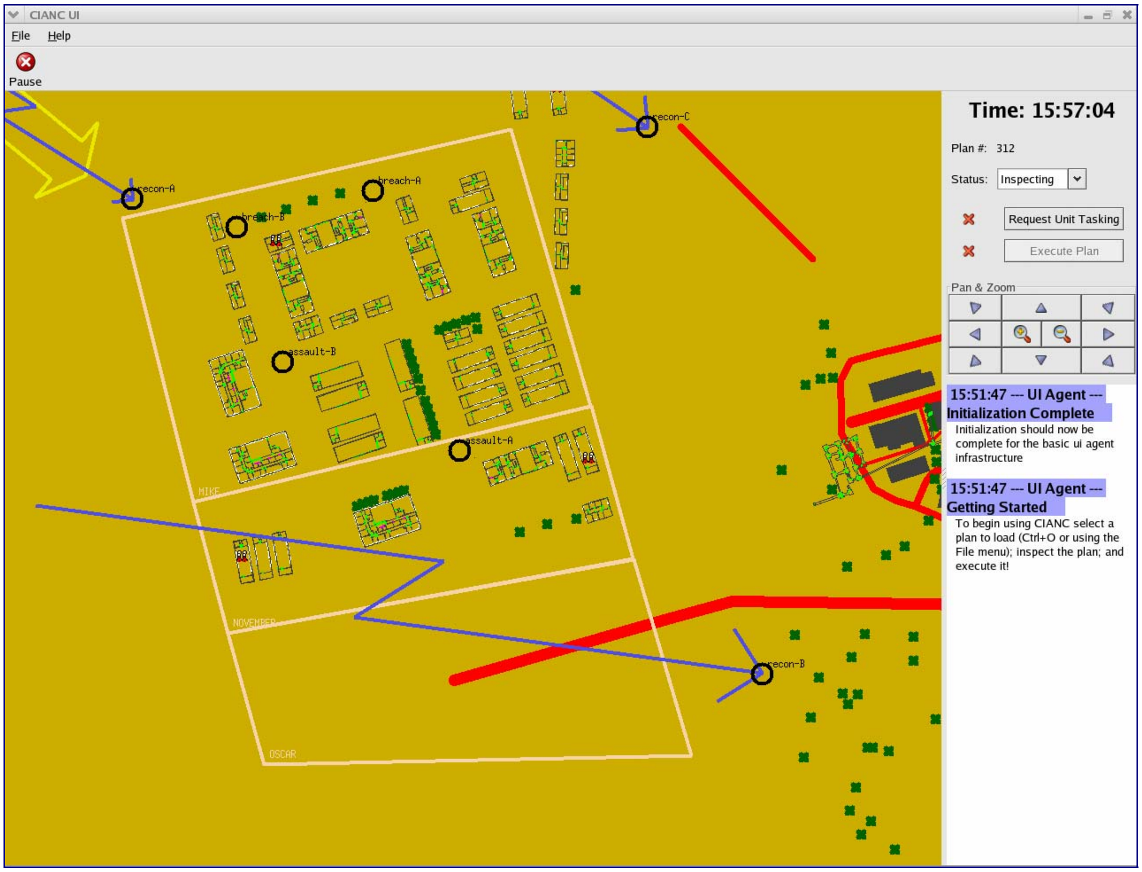


Figure 5. Map Display screen.

The second screen, the Plan Display (Figure 6) was designed around four information panes: objectives, decision points, points, and units. The objectives pane is used to display all of the mission objectives as specified in the OPORD. As objectives are completed, the list items status is changed as an indicator for the user. The Decision Points pane lists all of the decision points necessary to complete the objectives. For each decision point listed, the criteria for making a decision is indicated, and branch points are described if the decision cannot be made positively. As the user makes a decision, he or she tells the system to either continue with the mission, branch to a contingency plan, or halt the mission completely by pressing the appropriate check box. The Points pane is a list of waypoints used for mission planning. The Units panes are information only panes that allow the user to see the status and composition of all subordinate forces.

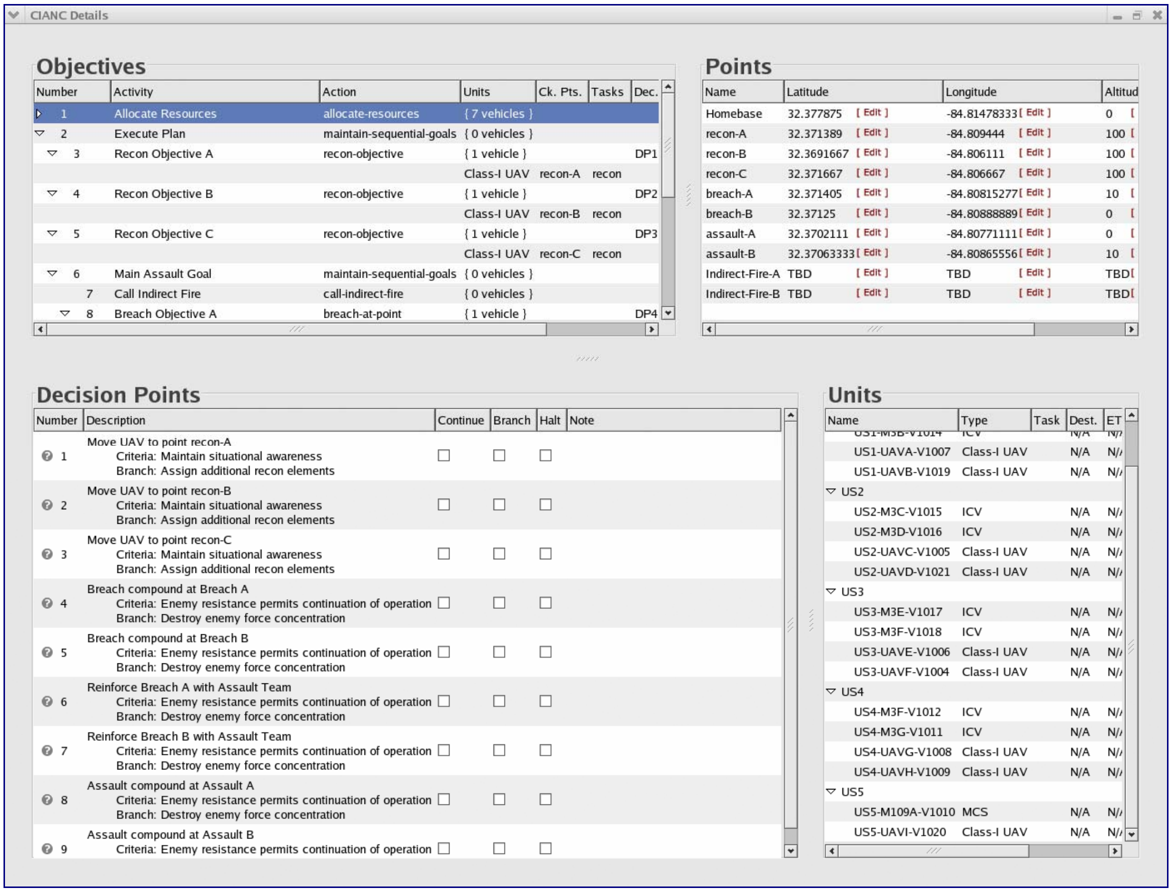


Figure 6. Plan Display screen.

System and Communication Architecture

The current CIANC[3] prototype integrates Soar-based interface agents into a combined simulation and operational environment for robotic control. The agents communicate using a FIPA compatible agent communication language (ACL), and a user interface to the agent processes was created using Tool Command Language (TCL) and Java. The main goal of the system was to allow a single operator/commander to better control/command multiple FCS robotic entities.

System Architecture

Figure 7 shows a component view of the CIANC[3] system architecture. Soar-based interface agents are integrated with an existing simulation system (either JSAF, Joint Semi-Automated Forces or OTB, OneSAF Testbed Baseline) via the CoABS (Control of Agent-Based Systems) grid. The user interface is built on top of the simulation system and communicates with the Soar agent application. Agents within the agent application can control and manage simulated robotic entities (task frames), and can communicate directly with the user interface.

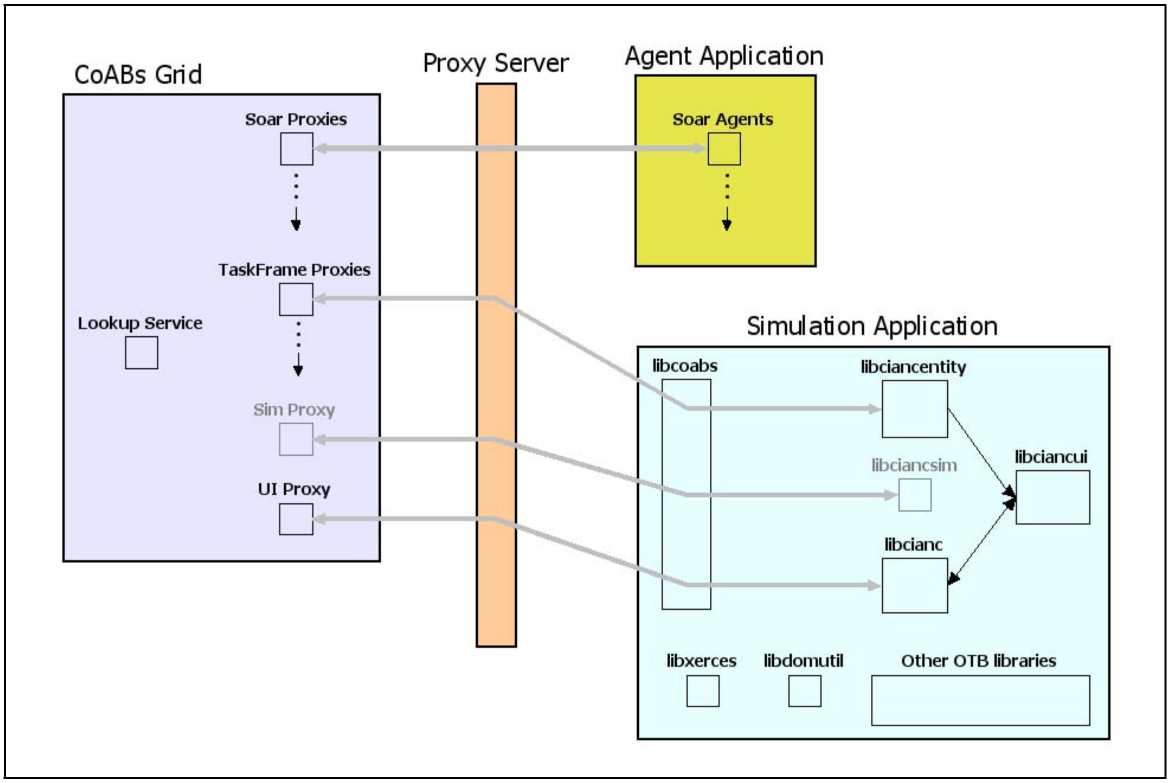


Figure 7. CIANC[3] system architecture.

The CIANC[3] system consists of three main components:

- Simulation Application – This application (either OTB or JSAF) is responsible for executing the underlying simulation. The CIANC[3] research team has added libraries to handle the communication between the Soar agents and the Task Frames. This application is currently also responsible for end-user GUI.

- Proxy Server – This server provides proxies for the Soar agents and the task frames to allow them to communicate over the CoABS grid. Proxies provide a consistent interface layer for heterogeneous agents and services to interoperate within the CoABS environment. The grid provides lookup services, logging and other agent management facilities.

- Agent Application – This application manages the run-time environment for the Soar agents and provides communication channels that allow them to communicate over the CoABS grid through their proxies. The agent application is built on the Soar cognitive architecture, which provides a scalable real-time reasoning and problem-solving environment.

All communication between these components is done using SoarComm (the Soar communication component) with Extensible Markup Language (XML) formatted messages. Any component that can send or receive messages is represented with a proxy on the CoABS grid (the simulation proxy might receive messages to pause or start the simulation for example).

The agent environment and communications framework are discussed in greater detail in the following sections.

Agent Environment: The Soar Cognitive Architecture

The Soar cognitive architecture is a powerful framework for creating multi-agent systems. It has been successfully used to model agents in various domains in complex battlefield simulations. Soar was used to create synthetic agents for FWA (Fixed Wing Aircraft), Rotary Wing Aircraft (RWA), related controllers, and more recently to model ground forces (Taylor, Koss, Frank, Nielsen, & Paul, 2001). For example, there are Soar models of fighters and strikers that interact with Soar forward air controllers during close-air support simulations. Similarly, for defensive-counter air (DCA) missions, Soar-based fighters coordinate with a Soar-based Airborne Early Warning (AEW) agent (currently in a simulated E-2C) that provides broadcast and close control support to fighters. In all cases, human operators can also provide command and control to Soar agents. This intervention is allowed but not required.

Agent Communications

Robotic forces must be able to communicate with each other in order to conduct joint operations. An agent communication language (ACL) provides a common way for agents to communicate. An effective ACL must enable interface agents to communicate between multiple echelon hierarchies of both robotic and human forces. A number of research groups have defined an agent communication language that can enable robotic forces to perform these types of communication, but the one considered most applicable is that based on Joint Intention (JI) theory (Cohen & Levesque, 1990c; Huber, Kumar, Cohen, & McGee, 2001). The JI ACL also offers several additional benefits. The JI ACL provides a formal semantics that allows interface agents to deal with actions explicitly. This would enable robotic forces to make decisions, maintain situation awareness and share information more efficiently. By using a JI-based ACL in the next generation of FCS, robotic forces would be able to execute commands rapidly and describe their actions precisely. Robotic forces would also be able to share awareness information about their current situation, status, plans and experiences. This would allow groups of robotic forces to coordinate activity.

CIANC^3 Agent Roles and Responsibilities

The CIANC^3 framework of cooperative interface agents is based on the roles found in current command staffs. Command staffs commonly provide the five basic functions mentioned earlier in this document to commanders in support of reconnaissance, security, offensive, and defensive operations (e.g. DA, 2003). These functions are: provide timely and accurate information; anticipate requirements and prepare estimates; determine courses of action and make recommendations; prepare plans and orders; and supervise execution of decisions.

In CIANC^3 these functions are divided between three classes of agents: Tasking, Monitoring and Coordinating which align with command, control and communication respectively. The idea is that interface agents form a layer between warfighters and battle command systems, and form ties between echelons and within echelons. Although other configurations are possible, the basic roles and responsibilities required of the interface agents needed for FCS remain. In addition, it is assumed that interface agents will have access to, and be tightly integrated with, other battlefield information and decision support systems. Regardless of the type of digitized services that will become available to battlefield commanders, the need for rapid tasking, coordinating and monitoring of operations will increase with FCS. These agent classes are discussed below with examples of how they might be used.

Tasking agent

Tasking agents are designed to assist commanders and controllers to rapidly issue battlefield commands. Ultimately, they would reason about the commander's intent, standard operating procedures, unit capabilities, operating environment and enemy disposition to present the commander with a reasonable operation plan. Where ambiguity exists, Tasking agents engage the commander in dialog to clarify intentions and present several options. After customizing the resulting plan as necessary, the commander can then issue the order. The Tasking agents then translate the order into the proper command sequences for next command layer. These sequences range from dialog completion information to atomic-level robotic commands, or relatively high-level commands that will be further processed by a cooperative planning system.

For example, a commander may wish to task a deployed company to attack a target. To do this he could select the company or individual platoon elements with a light pen (or other suitable input device) and drag them to the designated target area using the desired path and direction of attack. The Tasking agent would then query the commander as to the mission type who in turn would select some form of attack mission. The agent would then reason about the current posture of the company, assets of the platoon elements, terrain, weather and enemy, and propose a mission profile. An order would then be prepared specifying the commander's intent: movement orders indicating lead and screen elements, and other information normally included in an operation plan. After reviewing and verifying the plan, the commander would confirm the order. The Tasking agent would then translate the order (for robotic forces) and send out the plan. After confirming receipt of the order, the system would then monitor the plan's progress and update the commander as necessary.

It is not enough that the system simply automate the commander's tasks. Users of the system must be aware of and feel in control of the situation at all times. Otherwise, they will either lose trust in the system, reverting to manual control, or place too much faith in it, becoming complacent and jeopardizing lives. After orders have been issued, the plans are visible to the commander on the Phase II prototype (see Figures 5 and 6) so that they can be inspected, monitored, critiqued, and modified. This combination of interface agent assistance and direct manipulation is essential to achieving the right mix of automated and manual control. Examples of other Tasking agent work include:

- Tasking UAVs for targeting.
- Automatic weapon selection for known target types.
- Automatically modifying defensive posture in the event of an ambush.

- Modifying weapons usage (rate of fire, ammo selection).

- Modifying alert rules for when an autonomous agent should seek guidance.

- Facilitate any direct manipulation by providing context-sensitive assistance such as assigning targeting priorities.

Coordinating Agent

Coordinating agents are responsible for facilitating communication and coordination across and within echelons of the command hierarchy. While command hierarchies will certainly continue, operational hierarchies are likely to become more network-centric, blurring the distinction between separate commands. Units in one command may cooperate with a second command element one minute and a third the next. Such dynamic operational shifts will only be possible by automating much of the communication and coordination that must occur in such situations. Tasks such as determining radio frequencies, call signs, unit designations, chain-of-command, Identification, Friend or Foe (IFF) and communications security are all time-consuming but necessary work with which Coordinating agents can greatly assist.

More importantly, Coordinating agents can increase force lethality in cooperative engagements by minimizing duplication of effort, maximizing target coverage, synchronizing time of attack or massing fire on a single target. They can also be responsible for maintaining a common operational picture (and thus, situational awareness) by updating higher and lower echelons on the current situation, plans, enemy intentions and battle damage assessment. As with Tasking agents, it is important that Coordinating agent actions, processes and results be visible to the user. The commander must be able to verify that his intentions are being accurately implemented, and he must be able to intercede when necessary.

Another situation where coordination is critical is when responding to fast-moving or stealthy targets. It is often necessary to coordinate air defenses and sensor systems faster than humanly possible to effectively counter these attacks. In such situations, the Coordinating agent might work directly with Monitoring and Tasking agents to rapidly eliminate the threat. Other tasks that might be performed by Coordination agents include:

- Setting up direct sensor-to-shooter communications across commands.

- Setting up other cross-command tasking such as indirect fire support.

- Facilitating teleconferencing.

- Reestablishing communications and integrating orphaned units.

- Communicating routes, plans, intentions, progress and other explicit or implicit information.

- Sharing incomplete sensor information (such as vectors to fire source) to higher echelons.

- Facilitating direct control of vehicles (e.g., teleoperation) in critical situations.

Monitoring Agent

Monitoring agents are responsible for helping the commander maintain an accurate awareness of the current situation (situational awareness) at all times. The amount of information available to battlefield commanders will continue to increase to the point of informational overload. The main role of Monitoring agents will be to prevent information overload by fusing, filtering, and prioritizing raw data, and transforming that data into information that the commander can use in the context of the current situation. For example, different units may report different directional vectors for the source of sniper fire. The Monitoring agent could use this vector data to triangulate the sniper's position and recommend through the Tasking agent that indirect suppressing fire be called on that location. Another possible data fusion role could be more proactive. Monitoring agents could use templates based on intelligence formats (e.g., SALUTE reports, which specify the Size, Activity, Location, Unit, Time and Equipment of an observed enemy) to task sensors or prompt humans for missing fields.

Monitoring agents should also filter information to minimize distractions, especially when the commander is engaged in critical tasks. For example, if the commander is busy responding to an ambush on one unit, he probably doesn't care at the time that another unit's status is "Okay" and has not changed. Such routine status reports should be stored for future reference but kept in the background so as to not interfere with more important tasks. Likewise, such information can be prioritized by criticality or by relevance to the commander's current tasks. For instance, message traffic and information flow may increase dramatically during a firefight. Where loss of life or equipment is imminent, Monitoring agents could make relevant information that might prevent or mitigate the situation more salient for the commander (e.g., by color or ordering in a message list, or threat icons on a tactical display). Other Monitoring agent tasks might include:

- Automatically updating and synchronizing Common Operational Picture (COP) databases.
- Presenting appropriate data visually, such as unit location, direction, supply levels and damage status.
- Providing all messages relating to a single friendly or enemy unit to help build a broader picture from single events.
- Represent visually direct communication lines between shooters and sensors.
- Monitoring health and stress levels of human subordinates.

Specialist Agents

In addition to the more general agents that apply to any organization of a multi-agent team, the research team has developed an initial set of specialist agent types that are instantiated and applied for specific tasks.

Networked Effects Agents. Networked Effects agents respond to effects employment requests by matching the best effects delivery platforms to each corresponding request. This matching process includes: determining which battlefield platforms are available to be employed; querying ontologies to build an inference-based understanding of platforms, weapon systems, and targets; determining the feasibility of employing particular platforms and weapon systems against particular targets; employing requested effects against requested targets; and requesting maneuver of particular platforms and weapon systems into configurations more suitable to the employment of effects. By abstracting effects requests away from specifically identified platforms and weapon systems, the Networked Effects agents permit the formation of ad hoc teams on demand, reducing both kill-chain latency and commander workload overhead. In this way, Networked Effects agents can contribute significantly to increasing operational tempo for battlefield commanders.

Networked Sensor Agents. Parallel to Networked Effects agents, Networked Sensor agents respond to sensor information requests by matching the best sensor platforms to each corresponding area or target sense request. This matching process includes: determining which battlefield platforms are available to be employed; querying ontologies to build an inference-based understanding of platforms, sensor systems, areas of interest and targets; determining the feasibility of employing particular platforms and sensor systems against particular targets; employing sensors to obtain requested information; and requesting maneuver of particular platforms and sensor systems into configurations more suitable to the gathering of sensor information. As with Networked Effects agents, Networked Sense agents permit the formation of ad hoc teams on demand and increase the operational tempo for battlefield commanders.

Networked Maneuver Agents. Upon request, Networked Maneuver agents direct particular platforms to engage in maneuvers on the basis of platform capability descriptions. For example, a Networked Maneuver agent may request that a platform with an anti-tank capability and infrared (IR) sensing capability maneuver to a particular location (perhaps in response to a platform maneuver request generated by a Networked Effects or Networked Sense agent). When selecting the platforms, the Networked Maneuver agent will take into account the current tasking of particular platforms, the accessibility of platforms to the target maneuver location and the amount of time required for the platform to maneuver to the destination. This abstraction of maneuver requests away from specified platforms allows the fastest employment of the platform best matched to a particular request. Again, these features mean that Networked Maneuver agents can significantly increase the operational tempo for battlefield commanders.

Inter-Agent Communication Design

Robotic forces must be able to communicate with each other in order to conduct military operations. An ACL provides a common way for agents to communicate. An effective ACL must enable interface agents to communicate across multiple echelon hierarchies of both robotic and human forces. A number of research groups have defined an agent communication language that will enable robotic forces to perform this type of communication, but the most applicable is that based on Joint Intention (JI) theory (Cohen & Levesque, 1990c; Huber, Kumar, Cohen & McGee, 2001). The JI ACL also offers several additional benefits. The JI ACL provides a formal semantics that allows interface agents to deal with actions explicitly. This enables robotic forces to make decisions, maintain situation awareness and share information more efficiently.

By using a JI-based ACL, robotic forces will be able to execute commands rapidly and describe their actions precisely. Robotic forces will also be able to share awareness information about their current situation, status, plans and experiences. This will allow groups of robotic forces to coordinate activity.

The CIANC³ Agent Communication Language

Central to all interpersonal communication is the intent with which the communication is made and the interpretation of that intent by the recipient (Austin, 1962; Searle, 1969). In this speech act theory, the illocutionary force, or the intended result of the speaker, is differentiated from the perlocutionary force, or the actual result of the communication. The recipient of a message may interpret that message in different contexts, allowing the perlocutionary force to vary from that which was intended (e.g., the message sender may not be trusted and therefore the recipient may not believe the message). The mentalistic notions of beliefs, goals and intentions are quite naturally ascribed by humans to each other and to complex systems in general. It is this intentional stance (Dennett, 1987) that permits one to gauge the current state of the speaker and predict the future actions and state of the speaker. The intentional stance is particularly powerful when no other strategy works (e.g., physical stance, design stance). Agent communication languages are frequently defined in terms of the same mentalistic notions as that described by an intentional stance and therefore refer to the sender's and receiver's beliefs, goals, intentions, etc. The Soar agent architecture naturally supports ACL definitions. The ACL references to beliefs and goals are naturally mapped to Soar Working Memory Elements (WMEs) and goals, respectively. The mentalistic concept of intention (c.f., Bratman, 1987; Cohen & Levesque, 1990b) embodies a persistent commitment to act on a particular goal, which Soar also naturally captures in its operator execution framework.

This design used a variant of the Agent Communication Language semantics defined by Cohen and Levesque and extensions (Cohen & Levesque, 1990a; Cohen & Levesque, 1990b; Cohen & Levesque, 1990c; Cohen & Levesque, 1991a; Cohen & Levesque, 1991b; Cohen & Levesque, 1995; Huber et al., 2001; Kumar & Cohen, 2000; Kumar, Huber, Cohen & McGee, 2002; Smith, Cohen, Bradshaw, Greaves, & Holmback, 1998). The semantics were extended to included deontic modal operators.

Deontics

Deontic reasoning refers to thinking about which actions may, must, or must not be performed with respect to social/system norms. These conditions and limitations upon agent behavior are usually put into terms of permissions, obligations and prohibitions, respectively. Other deontic terms may be defined but are less common. For example, 'forbidden' is commonly a synonym for 'prohibited.'

In the study of deontics, the term Oxa (OBLIGATED x a), sometimes written Oa (OBLIGATED a) where x is left unspecified) says that agent *x* is obligated to perform action *a* and is taken to be a primitive in many formal theories of deontics (Von Wright, 1951; Horty, 1993; Jones & Sergot, 1996). The CIANC³ project formally ties this to the "Joint Intention" theory. By formally conjoining these two semantic theories, the following significant advantages are gained:

- A definition of what exactly the agents are obligated to do and the ramifications of the obligation. This is an important aspect of obligation and something often left undefined or vaguely expressed in the deontic literature. By basing the definition of obligation in terms of joint intentions, one can see that the agents are first required to perform an action (the ramifications of the obligation), then to reach mutual belief regarding success or failure.

- A specification of to whom the agent is obligated. While an agent may be thought of as becoming obligated to itself at some point in time (a form of intention, perhaps), most interesting for the CIANC^3 project are the obligations incurred between agents. Because of this, OBLIGATED is defined with respect to whom the agent is obligated, so the definition (OBLIGATED x y a) indicates agent *x* is obligated to do action *a* for agent *y*.

- A unified semantics that joins deep and rich intentional utterance semantics with the deontic aspects of obligations and permissions, which is further incorporated into a coherent specification of agent interaction patterns (communication protocols). Both semantics provide a key aspect of the full meaning in an utterance, but to this point the two aspects have not existed in a single cohesive, semantic framework.

In support of these claims the research team has defined a single, coherent set of basic semantic and notational definitions underlying joint intention theory. The JI definitions have changed semantically and notationally over time, and this can be confusing when piecing together a set of ACL performatives (communication acts or commands). A single, coherent set of performative definitions was defined. Prior research efforts led to narrowly focused redefinitions of performatives in the literature as the basic underlying definitions changed. However, not all performatives were updated with each underlying definition change, leaving a hodge-podge of sometimes incompatible or incongruent definitions. In addition, performative definitions have been modified over time even when the underlying semantic definitions have remained constant, ostensibly to remove limitations, provide extensions, etc. Finally, a broad, "complete" set of performative definitions was defined. Not all the performatives that might be considered necessary for fielding a multi-agent system had been previously defined in the literature, notably "utility" performatives implicitly required by joint intention theory, and those not so required but found to be useful when fielding systems based on ACLs and other semantics.

Agent Behaviors

The CIANC^3 prototype exercises two sets of basic capabilities: agent infrastructure capabilities, and tactical scenario capabilities. These capabilities were implemented using a combination of Soar agents and a domain ontology. An ontology is a formal knowledge representation of a particular domain that specifies objects, processes, relationships, concepts, and other entities. In this case, the domain ontology was used to model static objects in the military domain, such as vehicle and weapon types. Agent infrastructure capabilities include:

- Arbitrary sets of simulated Blue Force entities and their capabilities can be registered with, and accessed from, a prototype Directory Service.

- The Monitoring Agent can request, receive and propagate status messages from all entities registered with the directory services.

- The Tasking agent can dynamically assemble Blue Force teams based on the commander's plan requirements, and can establish system goals, subgoals and rules of engagement derived from the commander's plan.

- The Coordinating Agent can provide detailed instructions to Blue Forces, monitor for task completion or interruption, and react to plan interruptions.

The Phase II prototype is limited in its tactical reasoning abilities. At the current level of development, a small number of concrete exemplar scenario capabilities were created that highlight the range of future capabilities but do not necessarily reflect optimal tactics. Some specific tactical scenario abilities include:

- The system takes a general request for a UAV sensor platform to perform reconnaissance, and identifies and tasks specific assets.

- The system reacts to the loss of a UAV asset, noting the disruption of the plan and assigning a new asset to the task.

- The system assigns assets to routes, and issues fire requests and ROE (Rules of Engagement) changes.

Human-Agent Interaction Example

How the system assigns a UAV to a recon point is a good example of how the agent framework operates. The usage scenario involves coordinating an urban assault force of mixed human and robotic elements using an intelligent command and control system. The user, in a command vehicle, monitors multiple screens of the control system while making decisions and sending commands. Early objectives in the plan calls for the commander to conduct a reconnaissance of the objective area using unmanned aerial assets. Although the user issues commands and specifies recon points for the UAVs, most of the management of the specified task is accomplished by the agent system.

Figure 8 shows the general flow of control between human and the agent system when the user sends a command to conduct a reconnaissance of a specified objective. It is assumed that recon routes have been assigned during the Pre-Operation Phase planning. After the user initiates the recon action, a triggering event is sent to the Tasking agent. The Tasking agent then sends a message to the Directory Service within the CoABS grid, requesting an available asset to perform the task. The Directory Service identifies a specific UAV that is available and has the desired sensing capabilities. The Tasking agent requests approval from the user who then confirms the task with the chosen asset. The Tasking agent then sends the activated plan to the Coordinating Agent, informing it of the goal to recon an appropriate reconnaissance position with the specific asset. The Coordinating Agent then issues specific movement commands to the UAV. The UAV moves to position, sending status and sensor reports back to the Coordinating Agent via the Monitoring Agent. If the UAV is unable to complete the task, the Coordinating

Agent reports this to the Tasking agent, which then assigns a new asset (or informs the commander that there is a problem with the plan). Operation of the CIANC[3] interface is further detailed in the CIANC[3] Evaluation Training Manual.

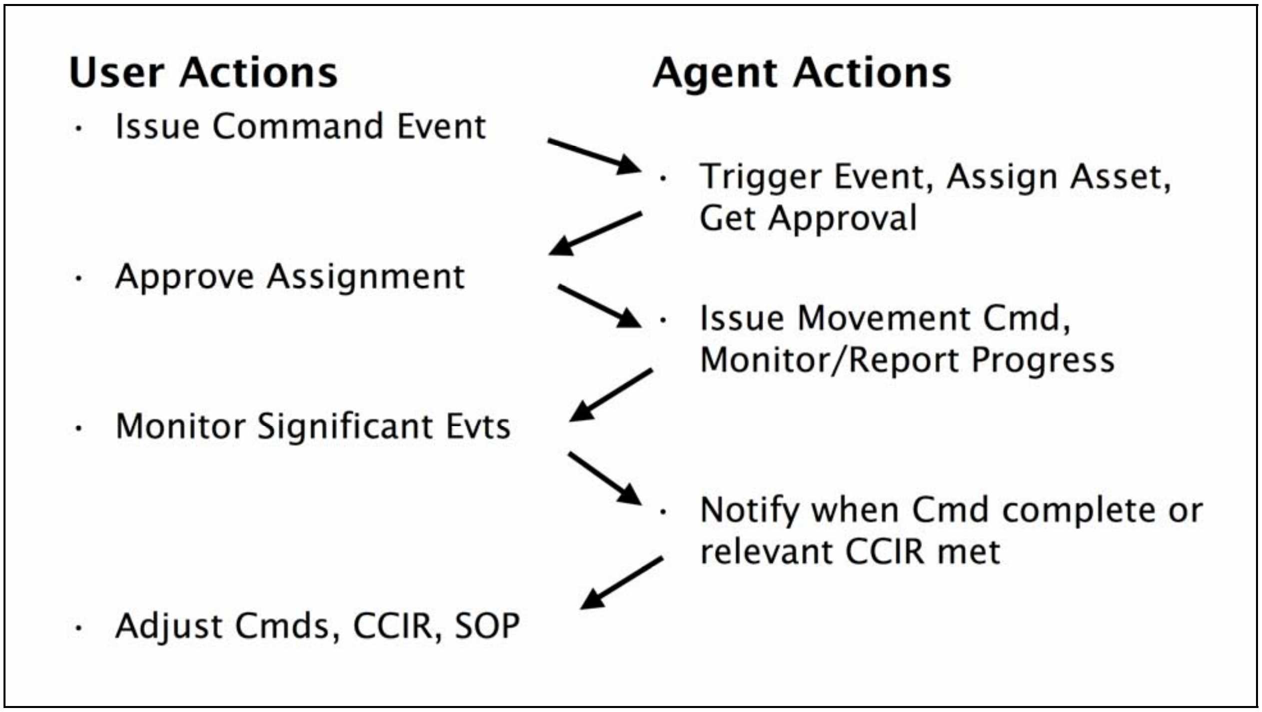


Figure 8. Human-agent interaction example.

While the implemented functionality represents a narrow slice through the problem space, the existing combination of basic infrastructure and scenario-specific capabilities demonstrate that an intelligent agent framework can be used to develop network sensing and effects, as well as policy-based maneuvering, while exhibiting rich domain knowledge, combined deliberative and reactive planning, and multi-level reasoning. This set of capabilities will be critical for the exploration and eventual fielding of supervisory command and control systems.

Ontology Integration

Currently, entire ontologies are represented in agent memory. Most existing ontologies remain modestly sized, and representing them directly in memory does not adversely impact performance in Soar. This solution also allows researchers to explore incremental transition of the ontology to long-term memory via Soar's native learning mechanism. The research team has implemented a translator, DAML2Soar, to map ontologies represented in DAML+OIL (DARPA Agent Markup Language plus Ontology Inference Language) into Soar agent run-time memory. The DAML2Soar generates a blackboard ontology representation that agent knowledge may use to retrieve class, property and relation information from the ontological knowledge base. The blackboard was designed so that the responses to these queries are cached once the initial response has been determined through deliberation. Responses are cached using Soar's native learning mechanism. The learned knowledge thus integrates the procedural domain knowledge with the declarative domain knowledge in the ontology. This approach deploys reusable components (agent architecture & learning mechanism, DAML+OIL ontologies, ontology reasoning knowledge) to realize agent knowledge bases optimized for speed and reusability. The DAML2Soar technology solution also facilitates experimentation to determine the limits of this approach and to explore alternatives to it.

Simulation Integration

Developing intelligent interface prototypes such as CIANC^3 requires integration with real and/or simulated data sources that can exercise the system and provide validity to the research. The current CIANC^3 system has been successfully integrated with the Joint SAF (JSAF, Joint Semi-Automated Forces) and OneSAF Testbed (OTB and OTB2) simulation systems. The integration provided CIANC^3 with simulated entities to control and receive status from, as well as a tool for creating opposing force behaviors, and a Soldier-in-the-loop research environment for formative evaluation and system refinement.

Phase II Usability Evaluation

The purpose of the evaluation was to assess the usability of the Phase II CIANC^3 prototype, particularly with respect to human interaction with the agent-based automation. The overall goal was to determine how and to what extent the research concepts and developed system components have the potential to reduce warfighter workload, reduce training requirements for human-robotic interaction, and improve mission effectiveness. The evaluation criteria were:

- Ability to successfully complete mission.
- Performance, such as task accuracy and completion time.
- Error rate, error type and error-inducing task methods.
- Situation awareness.
- Potential impact on mission performance.

Although the original plan was to evaluate the prototype interface more broadly with respect to overall usability, this was scaled back slightly to predominantly focus on functionality provided by the underlying agent system, and how this functionality could improve operator performance and effectiveness. As will be further discussed later in this section, the main change in the evaluation procedure involved the use of a “puckster” (a surrogate or assistant to input or implement the commands given by the user). This change in the evaluation procedure reduced participant training time significantly, enabling the evaluators to focus on the core question of whether intelligent user interfaces could help at a deep level, rather than be distracted by the more superficial implementation details of the prototype interface.

A total of nine active duty U.S. Army officers with training and experience at company-level operations participated in the evaluation. By design, all officers were either Captains or 1st Lieutenants to most closely match the intended user population. As indicated in Table 1, all were male and all but one had an Army Officer Area of Concentration (AOC) of 12A (Armor, General) or 19A (Armor). All had substantial experience participating in simulation exercises, but only half as computer system operators. Most of the participants reported using computers daily inside and outside work, but there was a wide range of game-playing experience for both training and entertainment. Only one participant listed any experience with simulated UAVs or UGVs.

Table 1

Participant Background and Experience

Participant	General Background			Computer Experience		Simulation Experience	
	Age	Rank	AOC	Training Games	Personal Games	Military Sims	Sims Used
P1	26-29	0-3	19A	Never	Never	No	
P2	34+	0-3	19A	Never	Some	Yes	JANUS
P3	26-29	0-2	19A	Never	Never	No	
P4	34+	0-2	12A	Some	Some	Yes	TACOPS
P5	34+	0-3	12A	Some	Daily	Yes	JANUS
P6	26-29	0-2	12A	Some	Some	No	
P7	30-34	0-3	19A	Never	Never	Yes	CCTT
P8	30-34	0-2	42A	Some	Some	No	
P9	30-34	0-3	12A	Some	Daily	Yes	JANUS, BBS

Note. TACOPS = Tactical Operations, CCTT = Close Combat Tactical Trainer, BBS = Brigade/Battalion Battle Simulation.

Apparatus

The Phase II CIANC[3] evaluation system consisted of two standard 1.5GHz PC's running standard Red Hat Enterprise Linux 3.0 Workstation operating systems, with two 19" CRT displays set at 1600 x 1200 resolution and 32-bit color. User commands were issued via standard three-button mouse and keyboard. The hardware was instrumented to collect user keystrokes and menu selections. Participant actions and speech were video-recorded from an over-shoulder angle as shown in Figure 1.

Data Collection Instruments

Several techniques were used to capture usability data and other relevant information. The objective for the use of multiple instruments was to seek convergence on key usability issues. Furthermore, since each technique addresses evaluation from a different perspective, using multiple instruments allows the collection of a broader set of data that should reveal more usability issues than any single technique alone. This was seen as especially important given the relatively sparse number of participants.

Background Questionnaire. A questionnaire was used to gather data relating to participant background (See 0). A key issue the questionnaire data was used to address was whether experience, training, or affinity for computer games affected participant performance with the CIANC^3 prototype. It was anticipated that participants with substantial experience with gaming and simulation would more readily accept computer automation, more easily grasp the skills necessary to complete the evaluation tasks, and would perform better than those without extensive computer experience.

Evaluator Observation. The CIANC^3 system was also evaluated according to the ability of participants to perform the evaluation task, the time it took to complete the task, and the type and severity of human errors and confusions experienced by participants. Evaluators observed each participant during performance of the conduct mission exercise, noting completion of tasks and apparent difficulties experienced. Participants were asked to perform using a "think aloud" protocol that helped evaluators infer usage concerns and difficulties. Actions and speech were recorded using two video cameras, and the user interface was instrumented to capture keystrokes. While these techniques supported measurement of task completion and difficulty, they did not provide a more objective measure of system capability in terms of users' situation awareness (Endsley, 1988).

Situation Awareness Global Assessment Technique (SAGAT). The SAGAT method (Endsley, 1995) was used to help assess the systems' ability to support users in attaining and maintaining situation awareness. The SAGAT technique provides a measure of situation awareness by comparing participants' perceived assessment of the situation with the actual situation. Measurement is accomplished by freezing the evaluation task at randomly selected times, suspending the simulation scenario, and blanking the user interface display screens while the participants answer questions about their understanding of the current situation. These perceptions are then compared to the actual situation based on system data or evaluation by a subject matter expert. If the performance interruptions are relatively few (three or less) and the duration of each SAGAT measurement is kept relatively short (5 minutes or less), the SAGAT method can provide a relatively unbiased assessment of participants' situation awareness without adversely affecting overall performance (Endsley, & Garland, 2000). The SAGAT questions used for this evaluation are provided in Appendix C.

Post-Evaluation Questionnaire. A post-evaluation questionnaire was administered to individual participants to gather subjective feedback regarding system functionality, ease of use, and other issues regarding system utility. Participants answered one set of questions by rating the system using a 7-point Likert scale. A second set of questions allowed the participants to write specific suggestions and comments regarding the system (See Appendix D for a complete list of questions).

Focus Group Questionnaire. A structured survey instrument was used to inform and guide group discussion for the final, focus group session (See Appendix E). Where the post-evaluation questionnaire was intended to gather feedback on the evaluation prototype that was developed, the Focus Group was to discuss how intelligent command and control tools might be used in the future, based on their prior experience and their experience with the CIANC^3 system. This instrument concentrated on soliciting battle command tasks and situations that were exceptionally cognitively demanding, such as maintaining situation awareness and synchronizing actions.

Procedure

Session Design. The evaluation process consisted of six sessions over the course of three days as shown in Table 2. The sessions were divided into three types and included: three single-participant sessions with think aloud protocols; two single-participant sessions with SAGAT; a final four-participant guided-discussion focus group.

Table 2

Evaluation Schedule by Session Type and Duration

	Day 1	Day 2	Day 3
9:00 -- 12:00	Think Aloud	SAGAT	Think Aloud
13:00 -- 16:00	Think Aloud	SAGAT	Focus Group

All sessions took approximately three hours to complete. The approximate schedule for each session was: 30-minute in-brief, 30-minutes of training, 60-minute conduct of mission, 30-minute survey and discussion, and 30-minute debrief. The key segments for each session segments were as follows:

- Participant completed background questionnaire to determine military experience in company-level operations and simulation software.
- Evaluator conducted in-brief to explain evaluation purpose and procedure.
- Military subject matter expert (SME) provided mission brief and rehearsal.
- Evaluator described and demonstrated system interface and functionality.
- Participant conducted mission and provided supporting data through Think Aloud or SAGAT measurement techniques.
- Evaluator served as “puckster” to perform system interactions during the mission, as directed by the participant.
- All participants completed post-evaluation questionnaire and group participants also completed the Focus Group questionnaire.
- Evaluator and military SME led debrief or group discussion.

During the single participant sessions, each participant completed the same mission twice, as discussed later. The SAGAT and think aloud measures were collected during the participant’s first conduct of the mission. One version of SAGAT questions (labeled SAGAT 1 in Appendix C) was administered at the start of the mission immediately after mission rehearsal and training. The second version (labeled SAGAT 2 in Appendix C) instrument was administered just after the participant had successfully initiated the breach, at approximately the

7-minute mark into the mission. For each administration, the prototype screens were blanked and the simulation suspended while the participant answered the SAGAT questions.

Immediately after both mission runs were complete, Think Aloud and SAGAT participants were asked to complete the post-evaluation questionnaire. Immediately after the completion of the questionnaire, the evaluation team reviewed participant responses to identify any additional interview questions based primarily on response outliers, as is common practice in usability evaluations. For example, since most responses were expected to be within a relatively narrow range of values, responses that indicated either exceptionally poor- or high-usability of some aspect of the system were further clarified. In some cases, such as when evaluators noted confusions during mission exercise execution, participants were asked to clarify what they were trying to do and what they were expecting the system to do. Such events indicate clear mismatches between system models and user models (e.g., expected system behavior).

During the group session, the four participants were divided into two 2-person teams. Each team conducted their missions separately while the other team was temporarily excused from the evaluation setting. The members of each team worked together as they completed their two repetitions of the mission exercise, using puckster assistance as with single-participant sessions. Group participants were asked to complete the Focus Group questionnaire immediately after the second team completed their last run through the mission exercise. Evaluators reviewed participant responses and prepared questions to lead the group through discussions. A single group discussion was conducted with both 2-person teams. Questions were asked about the relative importance of battlefield command tasks, which tasks were the most difficult to perform, how a CIANC^3-like system might be able to help in difficult situations, and what additional features and functionality would improve utility and likelihood of Soldier acceptance.

Evaluator Role. Evaluators recorded audio and video for all sessions with an emphasis on each participant's conduct of the mission and individual interview or group discussion. Evaluators observed participants conduct of the mission and took notes on participant behaviors, difficulties, questions, and comments. Evaluators timed performance tasks and elicited participant commentary when there were extended lulls in think aloud commentary or when participants appeared confused or frustrated. A typical evaluator elicitation took the form of, "What are you thinking about now?" Otherwise, evaluators did not interact with participants during the evaluation or help with system interactions unless it was requested by the participant or it was clear that mission progress had ceased.

Mission Tasks

The evaluation focused on the ability of each participant to use the CIANC^3 interface to conduct a simulated assault on an urban compound with an FCS company of predominantly unmanned systems. The participant was provided a pre-established mission plan and was responsible for executing the plan as quickly as possible. The mission, Commander's Critical Information Requirement (CCIR), and decision points were also pre-encoded into the prototype to reduce scenario complexity and to constrain participant actions. This was done to reduce participant training time and improve the ability to compare results across participants. To further constrain participant behavior, the simulated opposing force was intentionally restricted to a static, defensive posture. Additionally, the Map display always reflected simulation ground truth, so participants always saw the enemy forces (although they did not know that they were seeing all of the enemy). The mission entailed a set of key tasks and decision points with specified criteria for continuing to the next task. The mission tasks are listed in Table 3 and a detailed mission brief can be obtained from ARI.

Table 3

Usability Evaluation Mission Tasks

Tasks & Decision Points	Criteria
Recon Objectives A, B, C	Maintain SA
Call for Effects (indirect fire)	Resistance permits mission continuation
Breach Objectives A, B	Resistance permits mission continuation
Reinforce Objectives A, B	Resistance permits mission continuation
Assault Points A, B	Resistance permits mission continuation

The usage scenario described earlier to analyze how a warfighter might use the prototype system included several distinct phases including staging, pre-operation, operational, and post-operation. A detailed analysis of tasks by phase was performed for the evaluation's urban assault mission, and is available from ARI. Across all phases of the urban assault mission, a common set of generic tasks was developed that summarize the participants' performance requirements during the usability evaluation. The generic tasks are listed as follows:

- Use prototype to inspect and approve plan.
- Use prototype to request initial asset assignment.
- Evaluate assigned assets & asset routes.
- Approve assigned assets & asset routes.
- Use prototype to initiate battle sequence.
- Use displayed information and markers to maintain awareness of current battle progress.
- Interact with prototype to react to decision points as they arrive.
- Respond to prototype-generated CCIR notifications.
- Change the "Acceptable UAV Loss Ratio."
- Move recon points.
- Delete recon points.

Mission Brief and Rehearsal

A military subject matter expert provided each participant a mission brief and rehearsal to clarify mission requirements, tasks, constraints and success criteria. This brief addressed key and relevant aspects of the FCS unit the participant was to command and control with an emphasis on the capabilities and limitations of the unmanned systems and the network nature of FCS information and communication. A poster sized wall map of the mission setting (see Figure 9) was used to illustrate and rehearse mission tasks and decision points prior to using the CIANC[3] system. This wall poster was originally intended to provide the participant with a visual orientation and/or reference point.

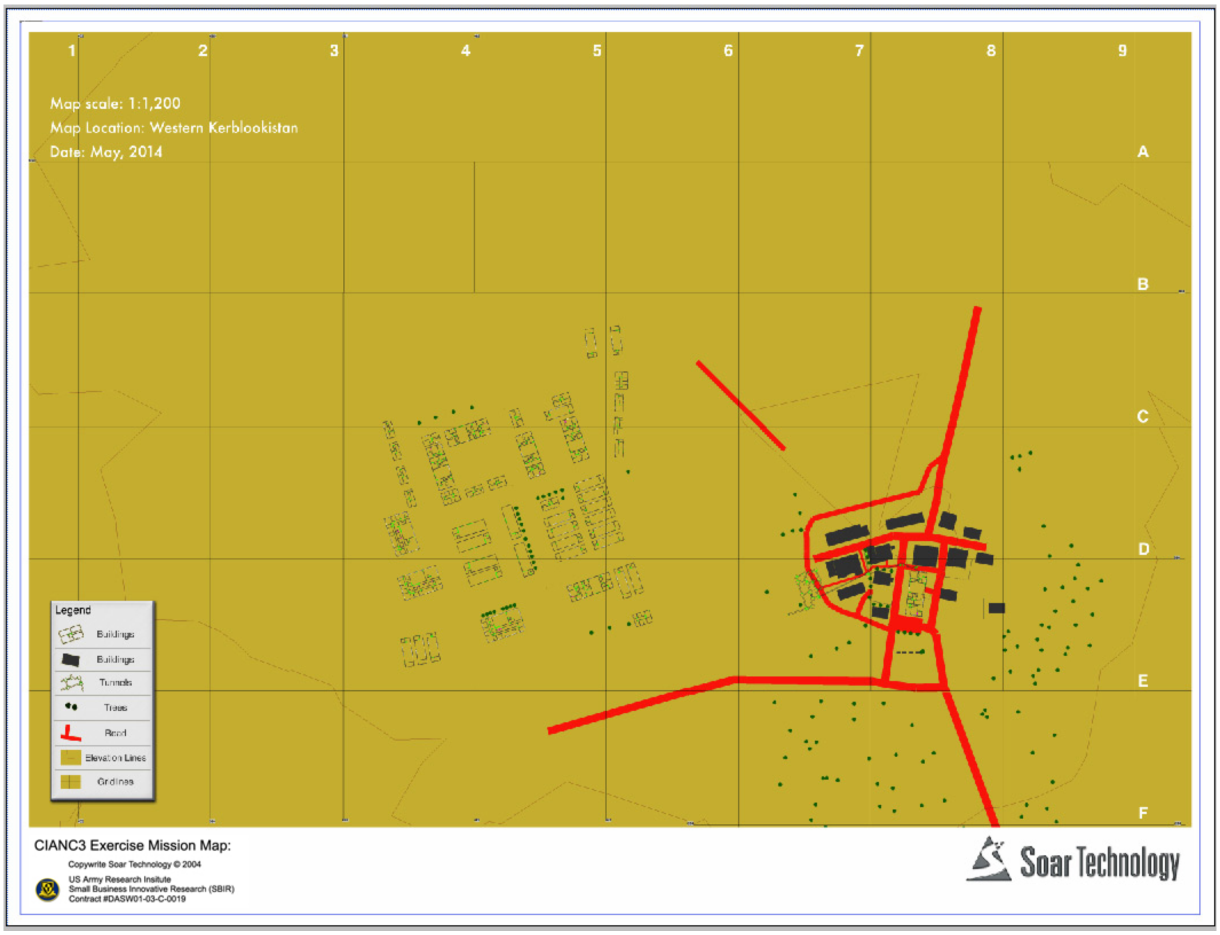


Figure 9. Wall map used for mission rehearsal.

Participant Training

After the mission brief and rehearsal, the evaluators provided scripted training to participants on the usage and functionality of the CIANC[3] user interface (see Appendix F). The training provided an introduction to the system that was read by an evaluator to each participant while seated in front of the user interface. During this familiarization training, interface features were demonstrated by the evaluator and performed by the participant with clarification provided by the evaluator, as requested. In addition, participants were provided a copy of the CIANC[3]

Training Manual (see Appendix F) as reference material available any time during the evaluation session. After the scripted introduction, participants completed a set of example tasks (see Appendix F) to assess their basic familiarization with the user interface. Participants were required to use the interface to find the information necessary to answer the questions correctly. Participants who had problems with the training questions were provided additional guidance from evaluators until all questions could be answered correctly.

Conduct of the Mission

During the mission exercise the participants were seated facing the two $CIANC^3$ display monitors, as shown in Figure 1. At the participant's side sat a supporting researcher who served as "puckster" to assist in performing computer interactions with the $CIANC^3$ system, as requested by the participant. Participants were given several reference sheets (provided in Appendix G) to assist them in conducting the mission and the evaluation tasks. Evaluators observed from behind the participant and recorded completion times for evaluation tasks and the mission and noted exceptional events, such as participant confusion or mistakes. Participants performed the same urban assault mission twice, as discussed below. Each mission was completed when all scripted tasks were performed and the mission objective was accomplished.

The use of an assistant puckster to help participants directly manipulate the $CIANC^3$ interface is a notable aspect of the evaluation procedures. The reasons for this assistance are discussed here and potential impacts on results are examined in the Discussion section. The primary reason for using a puckster was to focus the participant and the evaluation on the major concepts and functions represented by the interface agents rather than more minor and modifiable implementation issues. Participants faced a considerable challenge already in learning and employing the novel and complex FCS assets, particularly unmanned systems, provided for their urban assault mission. Requiring the participants to also acquire proficiency in manipulating the $CIANC^3$ interface would have increased the training load and perhaps impeded their ability to employ and assess more basic concepts and functions. It was also anticipated that the use of a puckster might increase the quantity and at best quality of each participant's verbalization of thought, intention, and action during the conduct of the mission exercise.

The procedure of having each participant complete the same urban assault mission twice also bears explanation. The primary rationale for mission repetition was to allow participants to spend the first trial better learning the mission and the $CIANC^3$ system and the second trial exploring alternate courses of action. Such repetition mimics a standard military training technique used for example in a Situational Training Exercise (STX) Lane that allows units to run the same scenario repeatedly to assess and explore different tactics and alternate courses of action. While multiple scenario runs are not standard practice for usability evaluations, issues that continue to surface even after experience tend to be more severe procedural errors that indicate a need for system refinement rather than training workarounds (e.g., Wood & Kieras, 2002).

After repeating the urban assault mission a second time, participants completed the remaining evaluation activities as previously described. These included completing the post-evaluation questionnaire, participating in a group discussion for the participants in the group condition, and receiving an evaluation debrief from an evaluator and military SME.

Results

Data were collected and analyzed for each of the three session types: Think Aloud, SAGAT, and Focus Group. All participants completed background questionnaires. Think Aloud and SAGAT participants also completed post-evaluation questionnaires and observations were noted regarding mission exercise execution. The SAGAT participants were also evaluated using the SAGAT instrument to assess situation awareness. Focus Group participants completed group questionnaires instead of the normal post-evaluation questionnaires. All groups participated in either individual debrief sessions or a group discussion.

Observation Results

The primary criterion of success for the evaluation task was whether the participant could successfully complete all mission tasks. The results were mostly positive with only one subject unable to complete the entire mission exercise. This participant was the only one with a non-Armor Area of Concentration and it seemed that most of the difficulty concerned tactics and decision-making rather than difficulty with the interface. All other participants completed the mission exercise in roughly the same time; the critical path was determined more by the simulation underlying the evaluation task scenario rather than by user actions. Other non-critical difficulties (e.g., those not affecting the ability to complete the task) were either observed by evaluators or taken from the think aloud protocol.

Usability issues were grouped into four areas: (a) automation design, (b) user interface design, (c) information design and, (d) miscellaneous. Although most participant interaction with automated aspects of the interface were positive, there was some confusion about why particular automations were happening, and some frustration at not being able to override the automated actions. Although these issues can mostly be seen as an artifact of how the CIANC[3] user interface was implemented, the ability for the automation to explain its actions and the capability for the human user to inspect and override any automated action seems to be a critical design feature for future development. Despite the difficulties, most participants wanted more automated-support rather than less.

Difficulties relating to the GUI design mostly centered on insufficient integration of display elements. For example, participants sometimes had difficulty relating information on the text-based Plan Display to graphical representations on the map-based Map Display. This was especially apparent when participants attempted to spatially relate the text-based decision-point information to a specific location on the map. Apart from these issues, participants reacted favorably to the information that was presented and how it was organized. Automated display of CCIR and other IR types was called out as being particularly helpful.

Issues relating to information design focused mainly on desired information that was not presented or information that was displayed in a non-standard or unfamiliar manner. For example, participants requested terrain information, structure elevations, line-of-sight information and other information that is typically combined from maps, photographs, human reports, and satellite imagery. They also wanted real-time data on items such as fuel status, ammunition available, unit capabilities and unit health. There were also difficulties with non-standard symbols and graphical controls used on the map display. In general, participants were pleased with automated display of sensor information and other intelligence information, but also wanted access to the raw sensor feeds and reports.

The other main issue, classified as miscellaneous, relates to the realism of the evaluation environment. Participants noted that in actual command situations, much time and effort is spent communicating information both to upper echelons and laterally to other commanders. Since such communication is a major source of cognitive and performance workload, they felt it difficult to accurately assess a system that did not consider that factor.

SAGAT Results

Table contains the results from the SAGAT evaluations. Overall the SAGAT results were positive and informative with 10 of the 16 questions answered correctly. There was a mix of situational awareness (SA) errors between the two participants. Both participants answered the following questions correctly:

- *Indicate the locations of each element on the map*. Although this result could have been influenced by planning and training time with map display, or by individual abilities to maintain an accurate mental representation of the battle area, this result also indicates that the participants were actively using the graphical display for problem solving and attending to the data presented as part of that display.
- *What do you expect the enemy to do in the next 5 minutes?* By maintaining a real-time view of the exercise battle area, and understanding the capabilities of the reconnaissance elements, participants were able to project their awareness of the current situation at least 5 minutes into the future. While this result may be influenced by the relatively straightforward mission scenario, having real-time intelligence information is likely a key enabler for accurate predictions of enemy behavior.
- *Which enemy element is your highest-level threat?* This indicates that enemy units are clearly indicated on the Map display and that the participants could differentiate between enemy unit types. It also indicates a consistent level of training regarding threat assessment (from prior combat training).

The one question that both participants answered incorrectly is a fairly important one: *How many casualties have you suffered?* As friendly unit assessment is a critical element of situation assessment for commanders, this result points to a need for better display of unit status and aggregate company strength. Although unit status is available, the necessary information is only available by selecting individual units from the Plan display.

Table 4

SAGAT Evaluation Results

Instrument SAGAT 1	Participant PC	Participant PE
Indicate location(s) of each element on the map.	Correct	Correct
Which of the following assets are available to support you?	Incorrect	Correct
Where are the principal enemy concentrations?	Correct	Incorrect
What do you expect the enemy to do in the next 5 minutes?	Correct	Correct
Instrument SAGAT 2	Participant PC	Participant PE
Which friendly forces are currently exposed to enemy fire?	Correct	Incorrect
Which enemy element is your highest-level threat?	Correct	Correct
How many casualties have you suffered?	Incorrect	Incorrect
Indicate which threats are currently under reconnaissance. Indicate those that are not.	Incorrect	Correct

Post-Evaluation Survey Results

Participants were asked to complete a survey and provide feedback regarding their use of the prototype interface. The survey consisted of 31 questions rated on a 7-point Likert scale and several free-response questions that allowed for discussion and comments (see Appendix D). Analysis of the post evaluation survey indicated considerable similarity between participant responses, as indicated in Table 5. The median score across all questions was 6, which indicated participants generally answered positively to all questions.

The focus of this type of early development-stage formative evaluation is not necessarily to confirm that the design was done correctly; it is assumed that good design must be an iterative process that depends on continual user input. While it is always good to get confirmation that a design is on the right track, the focus for this evaluation was to find design flaws and other usability issues within the system concepts and functionality that could compromise mission or prevent task completion. Perhaps more important was the second-order question of whether and to what extent did the application of intelligent agent technology contribute to successful or flawed system design. One important aspect of the survey results considered was the range of response values. When all responses for a particular question are uniformly negative or positive, the interpretation is typically clear. However, when the range of responses is wide, even if the median or mean value is within acceptable norms, it indicates that individual differences can play a significant role in the system's ability to support the task. While there will always be those who excel at particular tasks, one goal of good system design is to minimize the risk of complete task failures, irrespective of individual differences.

Questions where the median value was relatively high (at 6 or higher) and had a narrow range of values were considered to have a potentially positive impact on usability. Questions where the median value was below the overall median (lower than 6) and had a wide range of response values were considered to have a potentially negative impact on usability. For each question that fit in these categories, post-evaluation interviews sought to clarify the responses. These responses were then grouped into several categories and described further in the following section on usability findings.

Table 5

Post-Evaluation Questionnaire Results

Question	Participant									Range		
	P1	P2	P3	P4	P5	P6	P7	P8	P9	Low	High	Median
System Behavior												
… understandable	4	6	6	5	3	5	6	6	6	3	6	6
… predictable	5	5	6	6	5	5	6	6	7	5	7	6
… controllable	2	6	5	6	2	5	6	7	4	2	7	5
… appropriate	4	5	6	5	1	5	4	6	5	1	6	5
System Concepts												
… familiar	6	7	7	7	3	5	3	6	5	3	7	6
… extended well	6	7	6	6	4	5	4	6	5	4	7	6
System Terminology …												
familiar	4	7	5	6	4	5	3	6	6	3	7	5
… extended well	4	7	6	6	4	5	4	6	6	4	7	6
Work procedures												
… familiar	4	5	5	6	5	5	4	6	7	4	7	5
… extended well	4	5	6	6		5	4	6	7	4	7	5.5
System organization												
supported task	6	5	5	7	5	5	4	7	6	4	7	5
Information Display …												
clear	6	7	7	5	5	5	6	6	6	5	7	6
… sufficient	2	7	7	6	1	5	5	6	2	1	7	5
… relevant	3	6	7	6	6	5	5	6	5	3	7	6
… satisfying	5	6	7	5	5	3	5	6	5	3	7	5
Learning to operate …												
easy	7	7	6	7	7	NA	7	6	7	6	7	7
Controls												
… easy	7	6	6	6	7	NA	6	5	7	5	7	6
Locating functions &												
information easy	5	7	5	6	7	NA	6	5	7	5	7	6
System messages												
helped learning	4	6	6	6	2	5	6	5	5	2	6	5
Reference materials …												
clear	4	7	7	5	1	3	6	7	6	1	7	6
Training Time												
… sufficient	6	6	7	6	4	1	6	6	7	1	7	6
System Speed												
… fast	3	6	6	6	6	6	4	6	6	3	6	6
System Reliability												
… reliable	3	5	6	7	6	NA	6	6	6	3	7	6
Overall Reaction												
… positive	4	7	6	6	5	6	5	6	5	4	7	6
Using System												
… easy	6	7	7	7	6	6	5	5	6	5	7	6
… satisfying	5	7	6	7	5	6	5	5	6	5	7	6
… engaging	4	7	5	7	5	6	4	6	6	4	7	6
System												
… powerful	3	5	5	6	1	5	2	5	3	1	6	5
… flexible	2	6	6	6	1	4	4	2	2	1	6	4
… appropriate	4	7	6	6		6	4	6	4	4	7	6
… clear	5	7	6	7	3	6	4	5	6	3	7	6

Usability Findings from Post-Evaluation Survey

Survey questions whose responses indicated a large potential for effecting system usability were classified into five categories: (a) automation functionality, (b) automation implementation, (c) information design, (d) ease of use, and (e) ease of learning. These categories are further described in their respective sections.

Automation Functionality: The system needs to provide flexible and doctrinally correct automation. Automation functionality is defined here as the capability of the underlying agent-based automation system. Good automation functionality can often go unnoticed because it doesn't distract the user from their task. Poor automation functionality typically results in system behavior that is incorrect, unexpected, or undesired. In many cases, participants asked for more automation, but in some instances, they wanted more manual control over the automated system actions.

There were two specific concerns about system behavior that participants noted: automated behavior that was incorrect, and ability to override automated behavior. First, the way the system assigned units to objectives did not always seem correct to the participants. It also offered nothing to explain its behavior. This made the participants question the system's competence. This was a well-deserved criticism, since the logic the system used was limited and did not take into account a number of important criteria.

Second, the system's flexibility was limited in a manner that made it behave incorrectly in some circumstances. There are two primary examples of this. First, the participants were not able to control the movement of a number of their units. These units were pre-positioned and could not be moved. Second, the system would not allow the participant to substantially change the plan, either in terms of decision points or objectives. This led the system to follow a narrow set of steps that the participants would have changed if they had the opportunity.

Automation Implementation: The system needs to provide manual access to results of automation. Automation implementation is defined as the design of the user's interaction with the automation functionality (i.e., human-automation interaction). The category of automation implementation relates to how users were to interact with the underlying automation. This category is closely related to, and perhaps hard to distinguish from, the category of automation functionality. However, automation implementation relates more to the overall perception of how the automation fits within the user's task rather than specifics of what can be automated.

The participants gave low scores to the system's power and flexibility. Understanding that this was an evaluation prototype, the participants' reaction is not surprising. There were a number of features that the participants insisted were critical to system usage that were not present in the evaluated system. Most commonly requested features were:

- Ability to override automated task assignment and to manually assign units to objectives when appropriate.

- Ability to modify Decision Points: adding new ones, removing or editing existing ones, adding new branches and sequels as appropriate.

- Ability to modify Objectives, adding new ones, removing or editing existing ones.

Information design: The system needs to provide quick access to details to support aggregate information displays. Information design describes the integration of information elements within the CIANC[3] prototype graphical user interface. The category of information design was composed of a single question whose answers were substantially below median. This category relates to what information was available to make decisions and how it was presented.

Participants made a number of suggestions and design requests relating to information design, similar to evaluator observations noted earlier. The participants' general request was to provide additional data on the information display to improve situation awareness and decision-making. Specific requests that they made included:

- Display geographic locations of decision points on Map Display.
- Display all units on Map Display, including dismounts and UGVs.
- Improve clarity of unit identification and status displays.
- Improve use of standard graphics, and add legend for non-standard graphics.
- Add control for visualization layers, allowing commander to see different sets of details as needed, including terrain features.

Ease of Use: Task-centric design and application of automation contributed to user performance and system acceptance. Ease of use indicates that the procedures necessary to utilize the implemented functionality were straightforward to perform. It also indicates that the type and means of information that was displayed, was presented in a manner consistent with standard operating procedures. Specifically, the graphical information was displayed in a way that made perception and understanding natural for Soldiers trained to operate with paper-based geospatial artifacts. Additionally, the objective and decision point information corresponded well to the types of written orders and battle plans currently conducted primarily with non-digital methods. One participant noted that the objective and decision point information was very similar to information he currently manages by strapping a notepad to his thigh and updating it manually. System automation that is designed using task-centric or other user-centered techniques can have a strong impact on usability.

Ease of Learning: Designing to the user's mental model reduced learning time. Similar to ease of use, responses relating to ease of learning indicate strong congruence between CIANC[3] system design and the participants' mental model for maintaining situation awareness and making battle command decisions. In general, computer-based skill training requires a combination of procedural learning that maps computer procedures to operational needs as well as declarative knowledge that maps system-implementation concepts into operational concepts. Minimizing the amount of knowledge necessary to make such system-operational mappings can dramatically reduce learning time. Notably, the participants had a puckster to help manipulate the CIANC[3]controls. This meant that the participant did not necessarily need to learn the specific controls required for manipulating the interface elements. Although this discounts the high responses to some degree with respect to procedural learning, the results do support the contribution that matching system-implementation concepts and operational concepts can have on system usability.

Open-Ended Question Results

Participant comments to the open-ended questions on the Post-Evaluation Survey provided many useful suggestions for improving the $CIANC^3$ system as well as positive support for selected system concepts and functions. Features regarded as most useful included participants' strong endorsement of the $CIANC^3$ system's use of military schemas and a decision-centric approach to GUI design. As noted, the Plan Display (see Figure 6) presented users dedicated information panes for mission objectives and decision points. Participants reported that this aspect of the design provided the commander with relevant information in an understandable and actionable format that explicitly linked agents' activities with humans' decision-making processes. Participants also indicated this design provided an intuitive means for controlling subordinate activities at a more macro level.

One aspect of the $CIANC^3$ system that participants regarded with ambivalence was that of automatic tasking, such as re-tasking UAVs when one has been destroyed. Underlying their responses was a general concern that the pace and demands of future warfare (as characterized in the evaluation mission) would be difficult to manage without some assistance. Specifically, the idea of commanding unmanned systems in addition to human forces seems to increase workload on what is already a very demanding task. In this respect, the participants universally agreed that systems such as $CIANC^3$ would be welcome, if not essential for future warfare. What did not seem natural for participants was giving up control or trusting battlefield decisions and actions to a machine. While such concerns are normal, it should be noted that unconditional acceptance of battle command automation cannot be taken for granted. Furthermore, participants seemed to agree that systems that do provide task automation must be able to explain their actions or decisions when requested. Thus, participants simultaneously stressed the need for the assistance provided by the $CIANC^3$ system as well as the need to constrain that assistance to non-mission critical tasks.

Focus Group Results

The focus group discussion centered on how an intelligent battle-command system, as represented by the $CIANC^3$ prototype, might be used in a real-world environment. The written questions, and subsequent group discussion, were designed to guide participants along a chain of reasoning that included problem characteristics, problem definition, possible solutions, and ideas for further extensions and applications. Giving the participants some experience with the $CIANC^3$ system seemed to help solidify the abstract nature of intelligent automation systems and provided them with a command and concrete example on which to base discussion.

The participants' characterized situation awareness challenges mostly as expected. They described the challenges in terms of standard battle command tasks such as, determine enemy and friendly force location and status, assess and prioritize enemy threats and, in general, to "know the situation your Soldiers are facing." One unexpected challenge (given the nature of the mission scenario) was that of maintaining SA within a building or other structure. While this challenge seems specific to a narrow range of urban warfare missions, the problem can be generalized to any situation where visual and verbal cues are not available to the commander. This becomes very important to the design of future battle command systems where use of the system for SA may not be optional.

Participants also discussed situations and decisions where some form of automated assistance would be especially useful. The group initially focused on conditions when their normal cognitive abilities would be impaired, such as after conducting combat operations for multiple days with little sleep or rest. The group's desire however was not to give up control to the automated system, but rather to have the system help provide "sanity checks" on the decisions they would be making under stress and impairment. Other tasks discussed for which automation would be useful included clarification and display of adjacent friendly units and other available combat multipliers and assets. More mundane tasks included determining route feasibility, refuel point planning, and other logistical planning.

When asked to assess the value of automating specific activities, participant responses varied greatly. Several activities, however, were ranked highly by multiple participants. These included:

- Checkpoint placement – automate route planning, specify discrete points along the route for status checks and automatically processing and displaying status reports related to movement.

- CCIR and other reporting – automate and make explicit the linkage between observed world data and information requirements by superiors. Automate as much as possible the content of CCIR reporting and other situation reports.

- Movement and hazard-avoidance – automate the display of known hazards, obstacles, and alternate movement routes.

- Logistics and resupply – track fuel and ammunition levels and automate scheduling of resupply and maintenance, especially when commander is engaged in combat.

In general, the Focus Group participants were very supportive of the research pursued in this project and the prototype that was developed. Furthermore, they universally agreed that in the future much more automation would be useful and necessary. However, as with other participants, the Focus Group cautioned that too much automation, or poorly designed automation would be quickly rejected. Again these concerns seemed primarily focused on ability to control and predict system behavior, and to be able to inspect system reasoning when necessary.

Discussion

This section briefly summarizes the $CIANC^3$ system successes and issues associated with the Phase II evaluation. The issues identified based on participants' responses provide useful recommendations for refining the $CIANC^3$ system and adjusting the balance of human-machine control and functions in future development. The section closes with a discussion of how providing an assistant puckster may impact the evaluation of novel and complex systems.

System Successes and Refinement Issues

The CIANC^3 system as designed and evaluated showed a great deal of promise, but also exposed a number of critical issues with using agent-based technologies to support mixed teams of human and robotic forces on a dynamic battlefield.

Among the successes of the evaluation were:

- Demonstrated the ability of intelligent agents to control and coordinate robotic entities, allowing commanders to focus on higher level objectives.
- Demonstrated the use of intelligent agents to maintain and respond to changes in the battle-plan, helping the commander maintain SA and mission tempo.
- Demonstrated the use of a schema/decision-centric approach to GUI design, which presented the commander with relevant information in a form that linked the agent system with their decision-making processes.

As expected, there were some critical issues raised by the participants during the evaluation. In particular, these issues centered on the need to balance human-machine control and functions with a decided emphasis on machine support and human control. Based on this evaluation, critical system requirements for such systems should include:

- The commander to be able to override decisions at any time.
- A system with sufficient knowledge to produce doctrinally-correct suggestions.
- The system to be able to explain and justify suggestions.
- The commander to be able to reject and/or improve upon system suggestions.
- Visual thinking support.
- Complementary display forms to ‘snap together,’ highlighting common information across displays.
- The system to provide enough information for the commander to maintain SA and be able to confidently make his or her own decisions.
- Flexibility.

In sum, the evaluation was a positive step toward demonstrating that an intelligent agent system can support the commander’s management of human and robotic teams. Future CIANC^3 research efforts, however, must increase system competency, trustworthiness, and supervisory control. Refinements should also stress improving system flexibility so that commanders can more readily accept, modify, and improve system suggestions as well as better support the situational awareness of the commander.

Usability Evaluations for Complex Systems

Much of the literature and practice on usability testing assumes that the system being evaluated should be as close to "walk up and use" as possible. This is often not the case, however, particularly when the evaluation focuses on a novel and complex system such as a futuristic military command and control system. For such systems a substantial amount of training and experience is required to master the tactical and technical skills required to complete the mission and supporting tasks most germane to system objectives (Lickteig, et al. 2003).

As noted, the primary reason for using a puckster was to focus the participant and the evaluation on basic system concepts and functions rather than more minor and modifiable implementation issues. At this early stage of research and development there is more interest in how the warfighter reacts to the core functions of the system than in small usability details. It was also anticipated that the use of a puckster might increase the quantity and at best quality of each participant's verbalization of thought, intention, and action during the conduct of the mission exercise.

Post hoc, it seems that use of a puckster was a mixed blessing. It did help the warfighter focus on understanding and applying the CIANC[3] concepts and functions through all phases of the scripted mission and supporting tasks on successive trials. However, it also meant that the warfighter could multi-task more easily by assigning the puckster to interact with one task or operational concern while the warfighter moved on to assessing a different task or decision point. Such multi-tasking is inappropriate for a system intended for use by an individual warfighter. Having a puckster also may have altered think aloud verbalizations. Many of participant verbalizations requested specific system interactions by the puckster at a cost perhaps to verbalizations related to situation assessment and decision-making.

Method refinements might at least partially overcome some of the negative impacts of providing an assistant puckster. Probes or queries might be inserted during the mission requesting the participant to provide ongoing assessments of the situation to surrogate higher commanders. In addition, usability evaluators might develop methods for relating micro behaviors such as human-computer interactions, or participant requests for such interactions, to more macro command and control functions (see Lickteig, Sanders, Durlach, Lussier, & Carnahan, 2004). Most importantly, the greater the investment in participant training and experience early in system design and development, the greater the return on that investment.

Conclusions and Phase III Transition Efforts

The Phase II project was successful in many respects. This section summarizes the research and development conducted and the continued transition of the Phase II work to other military sponsored research efforts as well as prospects for additional commercialization.

A notable aside is the difficulty and delay experienced in the project to identify and access a suitable command and control platform with which to integrate the intelligent agent technology. Three alternative systems were analyzed, in depth, before ultimately foregoing the effort and developing an in-house simulated C^2 interface based on OTB. For various reasons (e.g., lack of functionality, non-supported code base, wrong target user), none of the candidate systems was deemed workable for the proposed research effort. Although the analysis did not produce a tangible by-product for the $CIANC^3$ prototype, much was learned in the effort, including how to determine integration needs for intelligent technologies, how to define the necessary communications and control mechanisms for such integrations, and the exact user population and technology gaps on which to ultimately focus. Such tangents are, of course, the nature of research and cannot be completely avoided. However, a need is identified here for an acceptable government-supported technology to support research and development in the area of battle command and C^2 systems. A long-term commitment to develop and maintain such technology for military sponsored research would go far to support the future research necessary to develop intelligent- and other advanced-technologies.

Summary of Phase II Results

Phase II of this project required research and development in a number of related areas, including artificial intelligence, software engineering, human-system interaction, and knowledge engineering. Much of this work either developed in a way to enable rapid transition, or was further matured by other-funded efforts because of a need for the core capabilities. Under Phase II of this project, major accomplishments included:

- Scenario Definition and Requirements Analysis – A scenario using an FCS-company to assault an enemy compound was developed and the commander's tasks and necessary decisions were analyzed to determine sufficient agent functionality. This scenario was used to project technological gaps that, if filled, would greatly improve user performance and effectiveness.

- System Architecture Design and Development – An architecture using the CoABS grid agent environment was designed to facilitate agent communications and human-agent interaction. No single technology, agent- or otherwise, is sufficient for overcoming the many challenges to be faced by future warfighters. Embracing an open-architecture approach to technology development improves the utility and applicability of new agent systems.

- Agent Communication Development – A FIPA-compliant communications protocol was developed to enable structured, well-defined communications between agents, humans and other system elements. Because no proposed GIG or other architecture can support multiple, heterogeneous agent types without a well-defined agent communication protocol, designing new agents with this assumption is key to future interoperability with other intelligent systems.

- Formal Agent-Interaction Protocol Definition – A formal deontic protocol was defined and implemented to simplify inherent agent design complexities, improve system robustness, and ensure verifiable agent system behavior. Military organizations are, perhaps, the epitome of deontic protocols: efficient organization and operation depend strongly on a well-defined structure of permissions, prohibitions, and responsibilities. It is likely that any intelligent technology that is fully integrated into military organizations must be capable of following orders just as every Soldier is taught. Such a capability greatly simplifies the development of cooperative multi-agent teams, as well as human-agent interaction.

- Ontology Research and Integration – A mechanism was invented to enable domain and doctrinal knowledge encoded within an ontology to be incorporated within Soar-based agents to simplify knowledge representation, maintenance and consistency. Knowledge representation has long been a key research area in artificial intelligence, software engineering, and philosophy. Representing "knowledge" in a way that can be developed, accessed, and reasoned about by different systems and for different questions, will be critical for building a large-scale knowledge-rich architecture such as the GIG. While the research performed under the CIANC[3] project only scratches the surface, it represents a necessary first step in addressing the knowledge problem by providing structural separation and interaction between agent-based reasoning systems and the knowledge on which they rely.

- Simulation Integration – The agent system and user interface were integrated with the OneSAF Testbed (OTB 2.0) to enable rapid development of realistic scenarios. Although OTB was not an ideal integration environment, it does represent a key Army simulation standard that supports research and development with its extensibility. It also enables further integration with other OTB and high level architecture (HLA)-compliant systems.

- User Interface Design and Development – A commander's interface was developed for the scenario that included a combination of a map-based display to support traditional spatially-oriented information and a task-oriented mission display for organizing information in a way designed to fit the objectives, tasks, and decisions necessary for the target user to perform the given mission. While the developed interface is not likely to be adopted as is, the methodology used to develop it demonstrates the commitment to maintaining a constant focus on users and their work in system development. The resulting design may represent a relatively useable and intuitive interface based on participant response.

- System Evaluation – A formative evaluation of the resulting system was conducted using U.S. Army Officers running a simulated scenario. Metrics included successful mission completion and ability to maintain situation awareness. Post-evaluation questionnaires and interviews were used to obtain additional feedback. The results were predominately positive, with most criticisms requesting more capability than could be developed within the scope of this project.

Overall, the evaluation demonstrated users' acceptance of intelligent support systems with some specific caveats:

- System must be capable enough to trust.

- Technology must behave in a predictable way.

- System must be able to explain its actions, and

- User must be able to override the system when necessary.

- Participants indicted they expect more of this supporting technology, with more capabilities and in more natural and useable forms. These results call for additional research to go beyond the contributions presented here – research that can address current limitations from multiple perspectives: human, technological, and military.

Research Implications

Much of the effort to date has gone towards creating the technical infrastructure that will permit more in-depth research into how intelligent interfaces can best be used by warfighters. This has resulted in a better understanding of how intelligent agents need to be designed and built for military applications, and how such agents can communicate and cooperate in synergistic agent teams. Specifically, the claim is that to the extent DoD applications will include the use of autonomous systems or services (agent-based or not), there must be a common and well-defined language for human-agent and agent-agent communications. Furthermore, depending on acceptable results to emerge from independently-designed systems is not good enough — there must be a rigorous definition of authority, permission, obligation and jointly-held goals for multi-agent systems to work.

Knowledge Representation and Use in Multi-Agent Systems

A stated goal of the U.S. Army is to greatly increase its warfighting effectiveness through the use of computer-augmented systems such as unmanned vehicles, intelligent interfaces, and command and control assistants. It is well understood that a significant increase in the autonomy, self-awareness, and configurability of these systems will be required if this goal is to be met. An important part of such autonomy and self-awareness is the ability to reason effectively over time, space and uncertainty. Performing such reasoning requires knowledge. A key challenge is how best to capture, encode, store, retrieve and reason over the knowledge. The claim here is that any highly capable system for assisting warfighters in battle command functions will need to solve this challenge in a general way. Once this challenge is solved for one area, such as operations, the resulting knowledge should readily apply to any number of related areas (such as training, planning, analysis, etc.).

An emerging requirement is that knowledge-based intelligent systems must be configurable by end-users—that is, by warfighters who are not familiar with artificial intelligence techniques or languages, and who cannot afford to be trained in these low-level details. As such, these systems must adjust their behavior in a way that is easy to understand and simple enough to do in a short timeframe.

Applicability of Knowledge-Intensive Intelligent Agents for Command and Control

According to Joint Vision 2020, military command and control will remain the primary integrating and coordinating function for operational capabilities and service components. To achieve this, Joint Vision 2020 goes on to explain, “Commanders will need a broad understanding of new operational capabilities and new (often highly automated) supporting tools in order to be capable of flexible, adaptive coordination and direction of both forces and sensors.” This requires that systems, at a minimum:

- Allow asynchronous object interaction.
- Provide messaging support for sporadic network connections.
- Provide richer peer-to-peer programming models.
- Provide secure communication with higher-level interfaces (Potok, Phillips, Pollock, & Loebl, 2003).

In their assessment of the needs of the FCS program, Potok et al., identify agent-based systems as the current or emerging technology that best meets those needs. In addition, it is believed that the objectives of Joint Vision 2020 and the nature of the military domain will also require that the agent-based system be knowledge-intensive (able to encode, access and reason over a large amount of knowledge) with a high degree of problem solving ability.

Primary goals of such an approach have been to work toward increasing the warfighter’s span of control for human-robot interaction and improving workload management. Current state-of-the-art involves multiple personnel controlling a single unmanned platform. The approach in this research centered on enabling a single person to control multiple unmanned platforms through mixed initiative monitoring of critical information requirements, delegation of platform control to intelligent autonomous agents, and ad hoc human and robotic team formation mediated by a multi-agent service-based architecture. Each aspect of the approach required developing agents that can reason over rich knowledge bases, including warfighter task models, weapon and sensor platform ontologies, COP blackboards, and sensor data streams.

Modeling the agent roles after human-staff Command, Control, Communications, Computers, Intelligence, Surveillance, Reconnaissance (C4ISR) roles, responsibilities and capabilities is central to this approach, and leverages the knowledge-rich character of the agents. To do this, the research team relies on the agents’ ability to access and reason over the knowledge sources listed above, as well as others. This approach to agent reasoning provides numerous benefits, including:

- Agent behavior that is more comprehensible and explainable to end users than are strictly analytic approaches.
- The ability to directly model agent problem solving on domain-proven solutions described in field manuals and doctrine.
- The ability to resolve issues of authority, responsibility and permission, which become ever more important with increasing autonomy, based on functional models that already exist in established command and control hierarchies.

Finally, by placing the question of knowledge representation and reasoning foremost, this research is taking steps toward a more unified approach to command and control systems. For example, with key knowledge repositories identified and formalized, the same multi-agent system can assist with information processing and robotic platform control for both commanding officers and robotic controller Non-Commissioned Officers. Knowledge-rich agents can reference shared knowledge repositories and provide different degrees of low-level control. In addition, having command and control systems based on knowledge-rich agents that are able to reference and reason over common knowledge bases will enhance command and control system development, enable the knowledge required by multiple sub-systems to be encapsulated and shared, and allow common agent capabilities to be used in multiple contexts.

Applicability of Knowledge-Intensive Intelligent Agents for Simulation Control

As called for by the Joint Vision 2020, simulation and experimentation will remain a key part of the innovation process. Using simulations to provide positive training, evaluation and valid experimentation requires that simulation controllers are able to develop and control large-scale scenarios with appropriate behavioral fidelity. This level of control creates workload and coordination challenges similar to battle command and could be served by similar supporting technologies.

The intelligent user interface approach described here has been developed, to this point, in simulation. While the eventual goal is to be able to transition the technologies to battlefield command systems, the approach has already begun to show its benefits in the simulation arena, providing the infrastructure for improved control mechanisms including: ad hoc group creation, multi-entity tasking, and entity- and group-level reactive planning and status monitoring. The basic infrastructure implemented here has been successfully integrated with a number of DoD simulation environments and promises to develop into a powerful tool for simulation control.

Commercialization and Transition Efforts

The area of research that has been addressed in this effort is highly relevant to many DoD and commercial needs. All indications are that technology will continue to expand at an ever-increasing pace, and the workload imposed on warfighters and other users has the potential to become overwhelming. The potential ill effects of overly complex technologies can be mitigated with the aid of intelligent user interfaces, especially those driven by teams of knowledge-intensive intelligent agents as described here. This work has resulted in numerous scientific and domain-specific publications (e.g. Wood, Zaientz, Beard, Frederiksen, Lisse, & Huber, 2004; Wray, Lisse, & Beard, 2004), and many aspects of this project have already been transitioned to other efforts or have been forged into separate research efforts. Example technology transfers include:

- Battlespace Information and Notification through Adaptive Heuristics (BINAH) – Office of Secretary of Defense (OSD)/Air Force Research Laboratory (AFRL)-ID Phase II SBIR to develop intelligent display and information delivery technology using CIANC^3-inspired agent teams.

- Robotic Command and Control Intelligent Enablers (ROCCIE) – Army CERDEC Phase II SBIR to research and develop technologies that will allow multiple, heterogeneous reasoning and knowledge systems to interoperate.

- Intelligent Control Framework – Army TARDEC Contract to develop technology for providing adjustable autonomy for CIANC^3-like intelligent assistant systems.

- Knowledge Enablers for Unit of Action – ARL Phase I SBIR to develop an Intelligence Agent to act as a specialist within the CIANC^3 framework.

- High-Level Symbolic Representation (HLSR) – Office of Naval Research project to research higher-level languages for programming intelligent agent systems, to simplify and speed development and improve system quality.

Conclusions

This report describes Phase II SBIR research efforts to develop agent-based intelligent user interfaces for battle command. Providing intelligent assistance at a level equal or greater to that of a human assistant requires large amounts of knowledge and a sophisticated reasoning system to apply that knowledge in real-time. The structure and design of the agent system described here is scaleable, malleable and rigorously well-defined. These techniques for defining and using the various forms of knowledge necessary for human-level reasoning will make future such development more inspectable, maintainable and verifiable. Finally, the type of communications and deontic framework developed through this research will be necessary for any robust multi-agent system.

The CIANC^3 system developed during Phase II allowed evaluation participants to successfully complete several clearly-defined performance tasks in a simulated FCS scenario. The feedback received from participants was supportive and constructive, providing an empirical base for further research and development. User satisfaction with the potential for the new system was demonstrated by a majority of user requests being for more of the types of automation provided by the CIANC^3 system. While some participants wanted more control and less automation, this was possibly due to intentionally limiting the complexity of the performance tasks. The prototype system appeared to be a good platform for training and performing unmanned asset management. Participants were able to effectively manipulate the position of multiple unmanned sensor assets in a way that maximized sensor coverage for the mission.

The most important results may prove to be Phase II advances in mixed-initiative technologies at the command, versus the operator, level. A triad of intelligent agents, Tasking, Coordinating, and Monitoring, has proven to be able to form the core of an intelligent user interface for command and control. These agents, which roughly correspond to command, control, and communications, respectively, can work as a virtual command staff for users to reduce workload and simplify complex tasks. Well-defined protocols for inter-agent communications were developed as well as the establishment of responsibilities, permissions, and prohibitions for those agents. Finally, this project proved to be a key enabler for future knowledge-rich intelligent systems via the development of bridge technology that connects ontologies with agent systems. Although these results are encouraging, future work should explore scalability issues; such as how cooperative agent clusters can operate and coordinate across echelons, in more complex scenarios, and under more realistic conditions.

In closing, the project supports the SBIR topic objective to develop intelligent interface agents for future battlefield commanders. The project examined and demonstrated many of the benefits to implementing such systems using knowledge-rich, intelligent interface-agents. Incremental results and technologies have already transitioned to other research areas including intelligence analysis, adjustable autonomy for unmanned system controllers, and agent and algorithm research for agent-based problem solving.

References

Alberts, D. S., Garstka, John J., Hayes, Richard E., & Signori, David A. (2001). *Understanding Information Age Warfare*: Command and Control Research Program.

Austin, J. L. (1962). *How to Do Things With Words*. Oxford, England: Oxford University Press.

Birmingham, W. P., D'Ambrosio, J. G., Darr, T., & Durfee, E. H. (1994). Coordinating Decision Making in Large Organizations. *Working Notes of the AAAI Spring Symposium on Computational Organization Design*.

Bradshaw, J. M., Dutfield, S., Benoit, P., & Woolley, J. D. (1997). KAoS: Toward an Industrial-Strength Open Distributed Agent Architecture. In J. M. Bradshaw (Ed.), *Software Agents*: AAAI/MIT Press.

Bratman, M. (1987). *Intentions, Plans, and Practical Reason*. Cambridge, MA: Harvard University Press.

Brinck, T., Gergle, D., Wood, S. D. (2001). *Usability for the Web: Designing Web Sites That Work*. San Francisco: Morgan Kaufmann.

Cohen, P. R., & Levesque, H. J. (1990a) . Rational Interaction as the Basis for Communication. In P. R. Cohen, J. Morgan & M. E. Pollack (Eds.), *Intentions in Communication* (pp. 221-256). Cambridge, MA: MIT Press.

Cohen, P. R., & Levesque, H. J. (1990b). Intention is Choice with Commitment. *Artificial Intelligence, 42*(3).

Cohen, P. R., & Levesque, H. J. (1990c). *Performatives in a Rationally based Speech Act Theory*. Paper presented at the 28th Annual Meeting of the Association for Computational Linguistics.

Cohen, P. R., & Levesque, H. J. (1991a). Confirmations and Joint Action. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence*. San Mateo, CA.

Cohen, P. R., & Levesque, H. J. (1991b). Teamwork. *Nous, 25*(4), 487-512.

Cohen, P. R., & Levesque, H. J. (1995). Communicative Actions for Artificial Agents. In *Proceedings of the First International Conference on Multi-Agent Systems* (pp. 65-72). San Francisco, CA.

Corkill, D. D. (1982). A Framework for Organizational Self-Design in Distributed Problem Solving Networks. Unpublished Ph.D., University of Massachusetts.

Decker, K., & Lesser, V. (1995, June) . *Designing a Family of Coordination Mechanisms.* Paper presented at the First International Conference on Multi-Agent Systems (ICMAS-95).

Dennett, D. C. (1987). *The Intentional Stance*: MIT Press.

Durfee, E. H., Kenny, P. G., & Kluge, K. C. (1998). Integrated Premission Planning and Execution for Unmanned. *Autonomous Robots, 5*, 97-110.

Endsley, M. R. (1988). *Design and evaluation for situation awareness enhancement.* Paper presented at the Human Factors Society 32nd Annual Meeting, Santa Monica, CA.

Endsley, M. R. (1995). *A taxonomy of situation awareness errors*. Lubbock, TX: Texas Tech University.

Endsley, M. R., Garland, D J. (Ed.). (2000). *Situation awareness analysis and measurement*. Mahwah, New Jersey: Lawrence Erlbaum Associates.

FIPA. (2000). FIPA ACL Specification. 2000, from http://www.fipa.org

Fox, M. (1979). Organization Structuring: Designing Large Complex Software, CMCS Technical Report 9-155 (Technical Report). Pittsburgh, PA: Carnegie-Mellon University.

Fox, M. S. (1988). An Organizational View of Distributed Systems. *SMC, 11*(1), 70-80.

Gasser, L., & Hill, R. W. (1990). *Engineering Coordinated Problem Solvers* (Vol. 4). Palo Alto, CA: Annual Reviews, Inc.

Horty, J. F. (1993). Deontic Logic as Founded in Nonmonotonic Logic. *Annals of Mathematics and Artificial Intelligence, 9*, 69-91.

Horvitz, E. (1999, May). *Principles of Mixed-Initiative User Interfaces.* Paper presented at the ACM Conference of Human Factors in Computing Systems, Pittsburgh, Pennsylvania.

Huber, M., Kumar, S., Cohen, P. R., & McGee, D. R. (2001). A Formal Semantics for Proxy Communicative Acts. In *Proceedings of Agent Theories, Architectures, and Languages (ATAL) 2001*. Seattle, WA.

Huber, M. J. (1999). *JAM: A BDI-theoretic Mobile Agent Architecture.* Paper presented at the Autonomous Agents, Seattle, WA.

Hutchins, E. (1995). *Cognition in the Wind*. Cambridge, Massachusetts: The M.I.T. Press.

Jones, A., & Sergot, M. (1996). A Formal Characterization of Institutionalized Power. *Journal of the IGPL, 4*(3), 429-445.

Kumar, S., & Cohen, P. (2000). Towards a Fault-Tolerant Multi-Agent System Architecture. Paper read at Fourth International Conference on Autonomous Agents, at Barcelona, Spain

Kumar, S., Cohen, P., & McGee, D. (2001). Towards a Formalism For Conversation Protocols Using Joint Intention Theory. *Submitted to Journal of Computational Intelligence*.

Kumar, S., Huber, M., Cohen, P., & McGee, D. (2002). Toward a Formalism for Conversation Protocols Using Joint Intention Theory. Computational Intelligence Journal (Special Issue on Agent Communication Language) 18 (2): 174-228.

Kumar, S., Huber, M. S., McGee, D. R., Cohen, P. R., & Levesque, H. J. (2000). Semantics of Agent Communication Languages for Group Interaction. *Proceedings of the 16th National Conference on Artificial Intelligence*, 42-47.

Labrou, Y. (1996). *Semantics for an Agent Communication Language.* Unpublished Ph.D., UMBC, Baltimore, MD.

Labrou, Y., & Finin, T. (1997). *Semantics and Conversations for an Agent Communication Language.* Paper presented at the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97), Nagoya, Japan.

Laurel, B. (1990). Interface Agents: Metaphors with Character. In B. Laurel (Ed.), *The Art of Human-Computer Interface Design* (pp. 347-354). Reading, MA: Addison-Wesley Publishing Company, Inc.

Leveson, N. G. (1995). *Safeware: system safety and computers*. Reading, MA: Addison-Wesley.

Lickteig, C. W., Sanders, W. R., Durlach, P. J., Lussier, J. W. & Carnahan, T. J. (2004). Human performance essential to battle command: Report on four future combat systems command and control (FCS C2) experiments (ARI Research Report 1812). Alexandria, VA: U.S. Army Research Institute for the Behavioral and Social Sciences.

Lickteig, C., Sanders, W., Lussier, J., & Sauer, G. (2003). Human Systems Integration for Future Command Systems: A Focus on Battle Command. *Proceedings of IITSEC 2003*.

Lieberman, H. (March 1997). *Autonomous Interface Agents.* Paper presented at the ACM Conference on Computers and Human Interaction [HCI-97], Atlanta, GA.

Maes, P. (1994). Agents that Reduce Work and Information Overload. *Communication of the ACM, 37*(7), 30-40.

Maes, P. (1998). Agents that Reduce Work and Information Overload. In M. T. Maybury, Wahlster, Wolfgang (Ed.), *Readings in Intelligent User Interfaces* (pp. 525-535). San Francisco: Morgan Kaufmann Publisher, Inc.

Malin, J. T., Schreckenghost, D. L., Woods, D. D., Potter, S. S., Johannesen, L., & Holloway, M. (1991). Making Intelligent Systems Team Players: Case Studies and Design Issues. *Volume 1: Human-computer Interaction Design; Volume 2: Fault Management System cases (NASA Technical Memorandum 104738).*

Maybury, M., & Wahlster, W. (Eds.). (1998). *Readings in Intelligent User Interfaces*: Morgan Kaufman Press.

Miller, P. (1986). Expert critiquing systems: Practice-based medical consultation by computer. New York: Springer-Verlag.

Miller, R. A., & Masarie Jr., F. E. (1990). The demise of the "Greek oracle" model for medical diagnostic systems. *Methods of Information in Medicine, 29*, 1-2.

Potok, T., Phillips, L., Pollock, R., & Loebl, A. (2003). Suitability of Agent Technology for Military Command and Control in the Future Combat System Environment. In *Proceedings of 8th ICCRTS CCRP*.

Rosenschein, J. (1985). *Rational Interaction: Cooperation among Intelligent Agents.* Stanford University, Palo Alto, CA.

Roth, E. M., Malin, J. T., & Schreckenghost, D. L. (1997). Paradigms for Intelligent Interface Design. In M. G. Helander, Landauer, Thomas K., Prabhu, Prasad V. (Ed.), *Handbook of Human-Computer Interaction* (Second ed.). Amsterdam: Elsevier.

Searle, J. (1970). *Speech Acts: An Essay in the Philosophy of Language*. Cambridge, MA: Cambridge University Press.

Searle, J. R. (1969). *Speech Acts*: Cambridge University Press.

Sheridan, T. B. (2000). Function Allocation: Algorithm, Alchemy, or Apostasy? *Int. J. Human-Computer Studies, 52*, 203-216.

Smith, I. A., Cohen, P. R., Bradshaw, J. M., Greaves, M., & Holmback, H. (1998). Designing Conversation Policies using Joint Intention Theory. In *Proceedings of the Third International Conference on MultiAgent Systems* (pp. 269-276). Paris, France: IEEE Press.

So, Y., & Durfee, E. H. (1994, March). *Modeling and Designing Computational Organizations.* Paper presented at the Working Notes of the AAI Spring Symposium on computational Organization Design.

So, Y., & Durfee, E. H. (1997). Designing Organizations for Computational Agents. In K. Carley, L. Gasser & M. Prietula (Eds.), *Computational Organization Theory*: AAAI Press.

Taylor, G., Koss, Frank, Nielsen, Paul. (2001). *Special Operations Forces IFORs*. Ann Arbor, Michigan: Soar Technology, Inc.

U.S. Department of the Army (2003) *Mission Command: Command and Control of Army Forces* (FM 6-0). Washington D.C.

U.S. Department of the Army (2005). FCS White Paper. Retrieved October 2005, from http://www.army.mil/fcs/whitepaper/FCSWhitePaper(v19_29Sep05).doc

Von Wright, G. H. (1951). Deontic Logic. *Mind, 60*, 1-15.

Wray, R. E., S. Lisse, & J. Beard. (2004). *Investigating Ontology Infrastructures for Execution-Oriented Autonomous Agents.* Paper read at 2004 AAAI Knowledge Representation and Ontology for Autonomous Systems Symposium, at Stanford, CA.

Wood, S. D. (1999). *The Application of GOMS to Error-Tolerant Design.* Paper presented at the 17^{th} International System Safety Conference, Orlando, FL.

Wood, S. D., Kieras, David E. (2002). Modeling Human Error for Experimentation, Training, and Error-Tolerant Design. Ann Arbor, MI.

Wood, S., J. Zaientz, J., Beard, R., Frederiksen, S., Lisse, & M. Huber. (2004). An Intelligent Interface-Agent Framework for Robotic Command and Control. Paper read at Command and Control Research and Technology Symposium, at San Diego, CA.

Wooldridge, M. (2000). *Reasoning About Rational Agents*. Cambridge, Massachusetts: The M.I.T. Press.

Appendix A

CIANC^3 Evaluation Training Manual

System Overview

The CIANC^3 system is designed to assist a U.S. Army company commander controlling multiple robotic entities. The current version of the software is designed to run in simulation, allowing the commander to direct simulated entities in a variety of missions. While controlling these missions, the commander is able to define relevant objectives and decision points, monitor the Common Operating Picture (COP) for current entity status, and make real-time updates to the current plan. The system supports this process by dynamically allocating units, as they are needed, monitoring status and resolving issues that do not affect the plan, and alerting the commander to plan progress.

COP Display

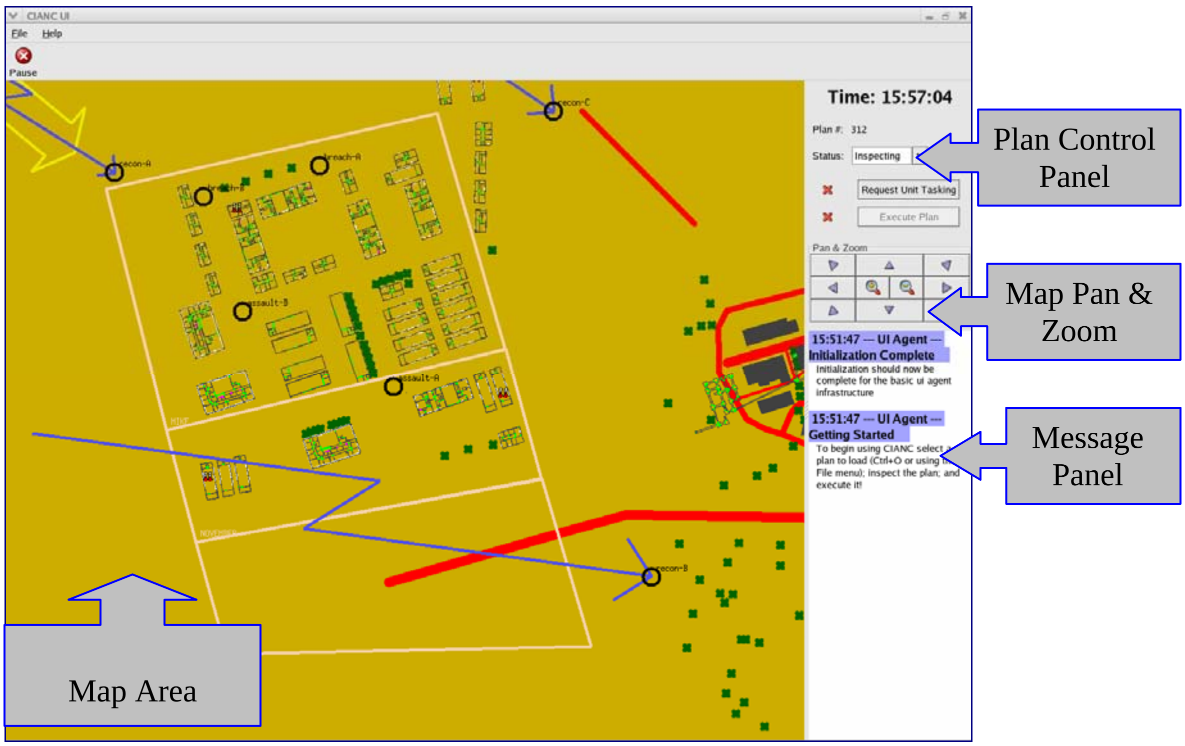


Figure A-1. COP Display. The Common Operational Picture Display provides a map-based view of the battlespace along with Map, Message, and Plan Control panels.

The COP Display provides the main Map, Message, and Plan Control panels. These panels provide the primary orientation to the current battlespace as a mission is executed. The COP provides a standard map based view of the battlespace, showing the positions and dispositions of friendly and enemy forces, checkpoints, routes and Named Areas of Interest (NAI), as well as terrain features such as buildings, roadways and vegetation. The Message Panel shows system-generated alerts, including unit task progress, commander critical information requirement (CCIR) notification, and decision point notifications. The Plan Control panel enables pre-operation management and control of the operation plan.

Plan Display

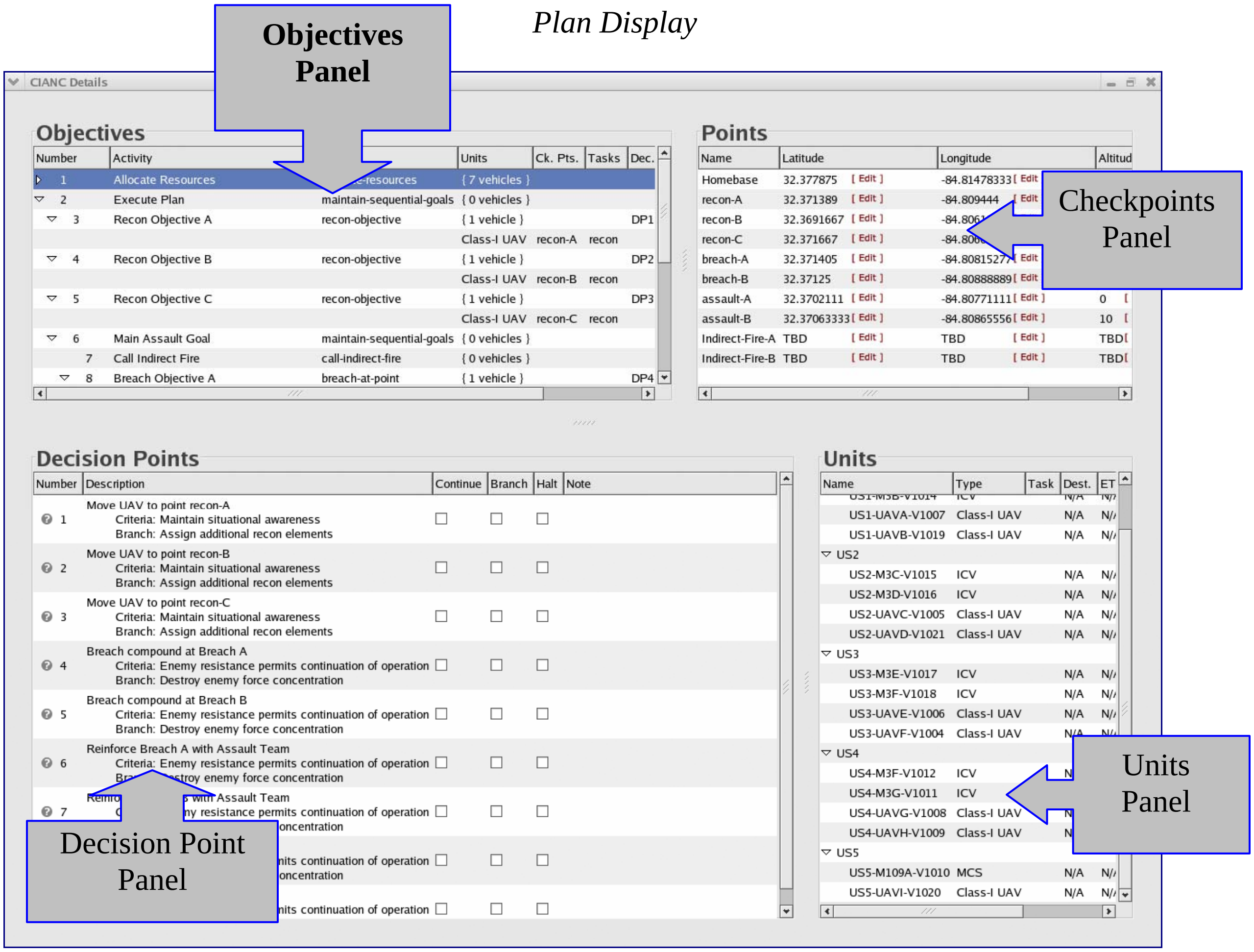


Figure A-2. Plan Display screen. The Plan Display provides detailed current information on plan decision points and objectives, as well as plan checkpoints and unit status via four display panels.

The Plan Display provides detailed access to the current plan decision points and objectives, as well as plan checkpoints and unit status. The Decision Points panel lists the current plan decision points and their status, allowing the commander to observe and control operation advancement. The Objectives panel shows specific named objectives and tasked units and their current actions toward the objective. The Checkpoints panel provides a listing of checkpoints identified in the current plan and their locations. The Units panel provides additional detail about simulated blue force units and their location and current disposition.

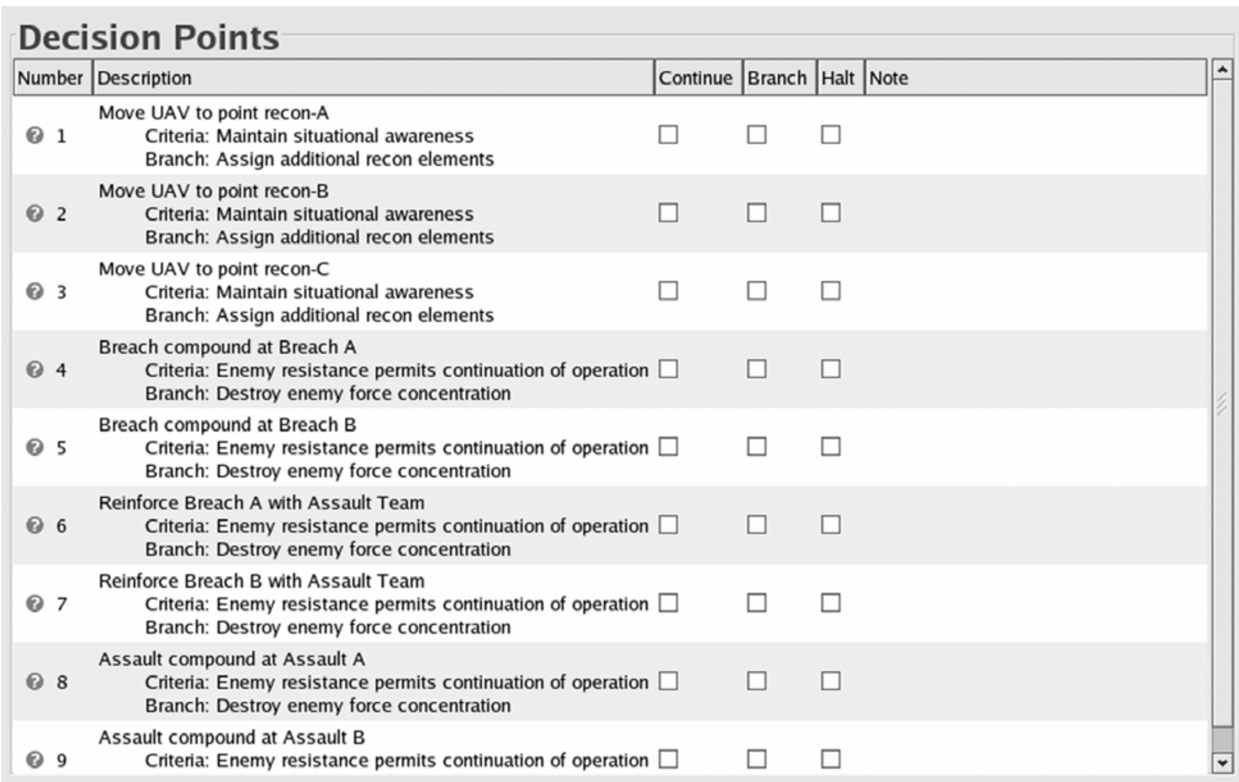
Method of Operation

Managing the Plan

A CIANC^3 plan is a computer representation of the current blue force Course of Action (COA), including decision points, objectives and CCIRs.

Decision Points

Decision points are defined as “the point in space and time where the commander or staff anticipates making a decision concerning a specific friendly course of action.” They include a set of military goals as well as specific decision-making criteria. During plan execution, the commander is notified when each decision point requires a decision to be made. Currently, the CIANC^3 system only allows the commander to authorize or reject plan continuation. The commander cannot direct the system to execute plan branches.



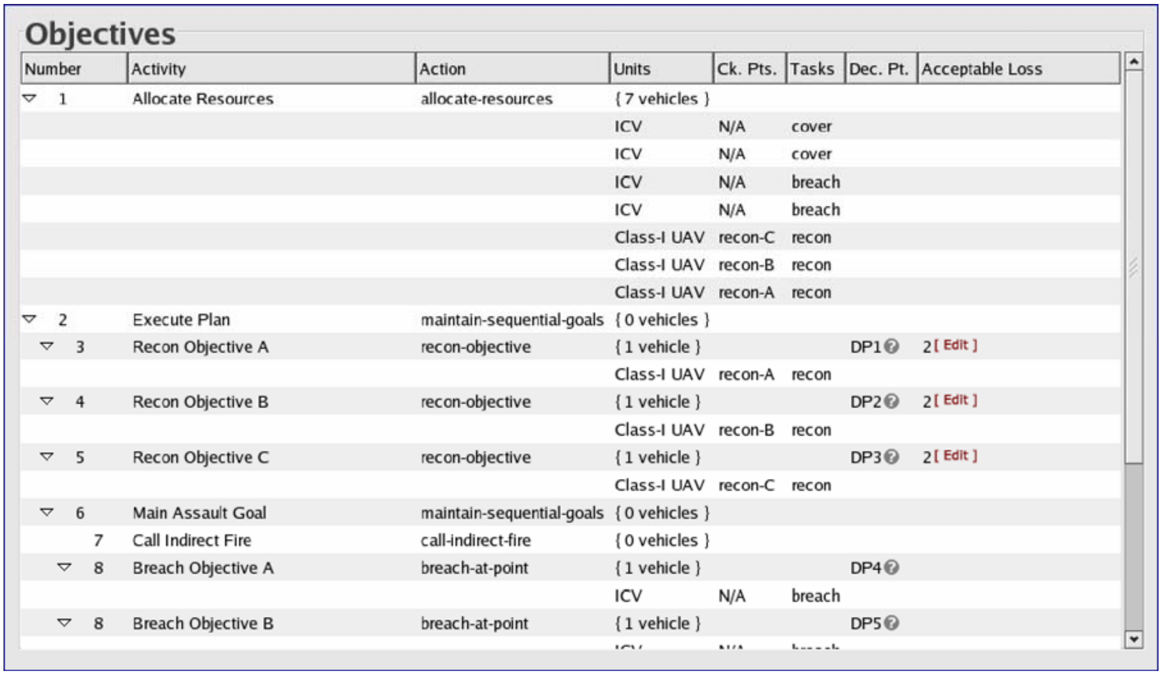
Decision Points

Number	Description	Continue	Branch	Halt	Note
1	Move UAV to point recon-A Criteria: Maintain situational awareness Branch: Assign additional recon elements	☐	☐	☐	
2	Move UAV to point recon-B Criteria: Maintain situational awareness Branch: Assign additional recon elements	☐	☐	☐	
3	Move UAV to point recon-C Criteria: Maintain situational awareness Branch: Assign additional recon elements	☐	☐	☐	
4	Breach compound at Breach A Criteria: Enemy resistance permits continuation of operation Branch: Destroy enemy force concentration	☐	☐	☐	
5	Breach compound at Breach B Criteria: Enemy resistance permits continuation of operation Branch: Destroy enemy force concentration	☐	☐	☐	
6	Reinforce Breach A with Assault Team Criteria: Enemy resistance permits continuation of operation Branch: Destroy enemy force concentration	☐	☐	☐	
7	Reinforce Breach B with Assault Team Criteria: Enemy resistance permits continuation of operation Branch: Destroy enemy force concentration	☐	☐	☐	
8	Assault compound at Assault A Criteria: Enemy resistance permits continuation of operation Branch: Destroy enemy force concentration	☐	☐	☐	
9	Assault compound at Assault B Criteria: Enemy resistance permits continuation of operation	☐	☐	☐	

Figure A-3. Decision Point panel. Decision points include a set of military goals as well as specific decision-making criteria.

Objectives

Objectives are defined as “the clearly defined, decisive, and attainable goals towards which every military operation should be directed.” They include the specific actions defined as essential to the commanders’ plan. The CIANC^3 system tracks the specific actions that are required to complete each objective, including the units assigned to the action, the tasks they are to complete and their current status.



Objectives

Number	Activity	Action	Units	Ck. Pts.	Tasks	Dec. Pt.	Acceptable Loss
1	Allocate Resources	allocate-resources	{ 7 vehicles }				
			ICV	N/A	cover		
			ICV	N/A	cover		
			ICV	N/A	breach		
			ICV	N/A	breach		
			Class-I UAV	recon-C	recon		
			Class-I UAV	recon-B	recon		
			Class-I UAV	recon-A	recon		
2	Execute Plan	maintain-sequential-goals	{ 0 vehicles }				
3	Recon Objective A	recon-objective	{ 1 vehicle }			DP1	2 [Edit]
			Class-I UAV	recon-A	recon		
4	Recon Objective B	recon-objective	{ 1 vehicle }			DP2	2 [Edit]
			Class-I UAV	recon-B	recon		
5	Recon Objective C	recon-objective	{ 1 vehicle }			DP3	2 [Edit]
			Class-I UAV	recon-C	recon		
6	Main Assault Goal	maintain-sequential-goals	{ 0 vehicles }				
7	Call Indirect Fire	call-indirect-fire	{ 0 vehicles }				
8	Breach Objective A	breach-at-point	{ 1 vehicle }			DP4	
			ICV	N/A	breach		
8	Breach Objective B	breach-at-point	{ 1 vehicle }			DP5	

Figure A-4. Objectives panel. Objectives include the specific actions defined as essential to the commanders' plan. The CIANC[3] system tracks the specific actions that are required to complete each objective, including the units assigned to the action, the tasks they are to complete and their current status.

CCIRs

The CCIRs are "a comprehensive list of information requirements identified by the commander as being critical in facilitating timely information management and the decision-making process that affect successful mission accomplishment." The CIANC[3] system can represent and, to a degree, manage each of these plan elements in support of the commander. Unlike decision points and objectives, the CIANC[3] Plan Display does not display the current CCIR list. Instead, the commander is notified about CCIR status changes via the COP Display messages.

Figure A-5. COP messages. Commander is notified about CCIR status changes via the COP Display messages.

In the current evaluation version of CIANC^3, the basic mission plan is provided to the commander and can only be changed in minor ways. (See Managing Decision Points and Managing Objectives below.)

Executing the Plan

To effect this support, at the direction of the commander the CIANC^3 system, identifies and tasks available units in manner consistent with the plan. Once the commander has reviewed and is satisfied with the plan and unit assignments, the commander may execute the plan. Executing the plan triggers the system to direct assigned units to complete the first set of objectives and to start monitoring the first decision points.

To execute the plan, the commander presses the ‘Execute Plan’ button located on the upper right corner of the COP Display. Once the button is pressed the system will immediately direct units to act.

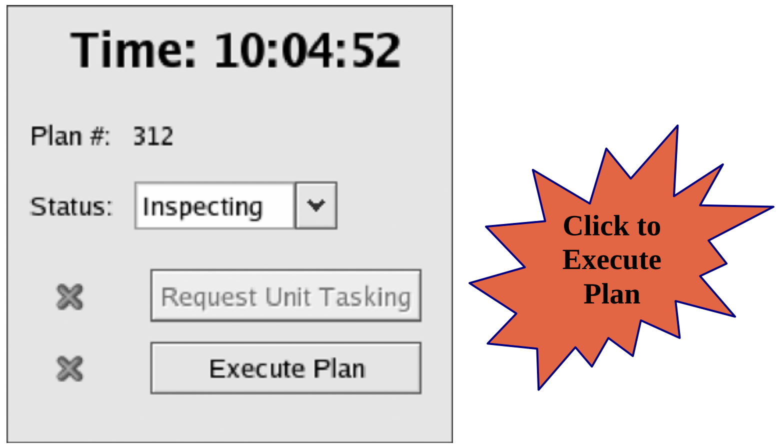


Figure A-6. Requesting Initial Unit Assignments to execute the plan, the commander presses the ‘Execute Plan’ button located on the upper right corner of the COP Display. Once the button is pressed the system will immediately direct units to act.

Managing Decision Points

During plan execution, the commander has limited control of decision points. The list of decision points is preset in the plan. Currently the commander cannot add or remove decision points. The commander’s primary responsibility is to understand the decision points and how they relate to the plan, to monitor the points to know when key plan decisions must be made, to make appropriate plan decisions, and finally to mark the decision points with the results of those decisions.

The CIANC^3 system progresses through the plan at a pace determined by the commander. The commander exercises control over plan execution by managing decision points. By marking decision points as being successfully completed, the commander authorizes future mission progress. Pending this approval, the CIANC^3 system will not have units initiate new action.

Decision Point Notification

The CIANC^3 will notify the commander when the system believes that a decision point has been reached, by highlighting the decision point and displaying “pending” in the Decision Point panel’s ‘Notes’ field.

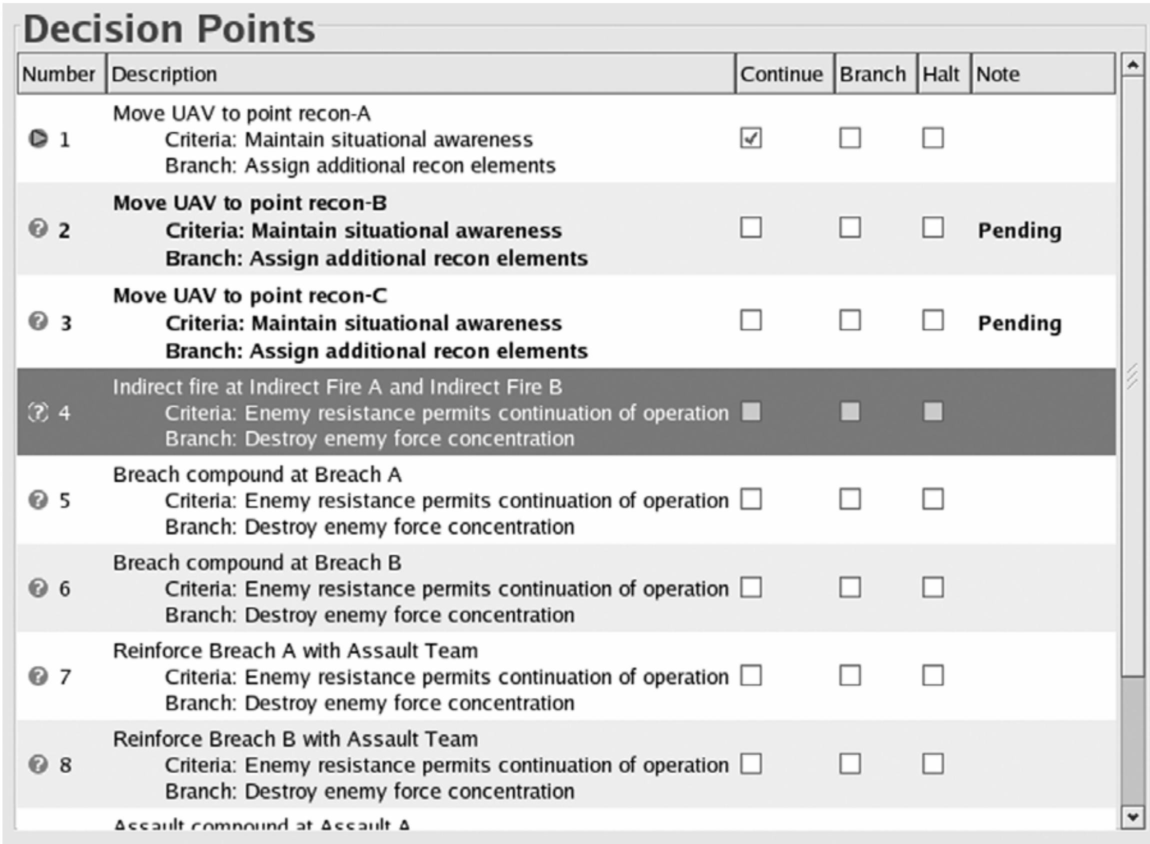


Figure A-7. Decision point notification. CIANC^3 will notify the commander when the system believes that a decision point has been reached, by highlighting the decision point and displaying “pending” in the Decision Point panel’s ‘Notes’ field.

Decision Point Status

CIANC^3 will mark decision points with the following status icons:

Undecided : Decision points begin as undecided and remain undecided until the commander marks the decision point to be continued or halted.

Continue : Decision points are given the ‘Continue’ arrow when a commander marks a decision point to be continued.

Abort : Decision points are given the ‘Halt’ X when a commander marks a decision point to be halted.

At any time, including before the decision point has been reached, the commander may mark that the decision point is complete and that the mission should continue. By not marking the point as complete, the commander is instructing the system to wait for approval before commencing further actions.

Marking Decision Points as Complete

The commander marks a decision point as complete by clicking the point's 'Continue' checkbox, as in the example below. In this evaluation, the commander may not select the 'Branch' or 'Halt' checkboxes.

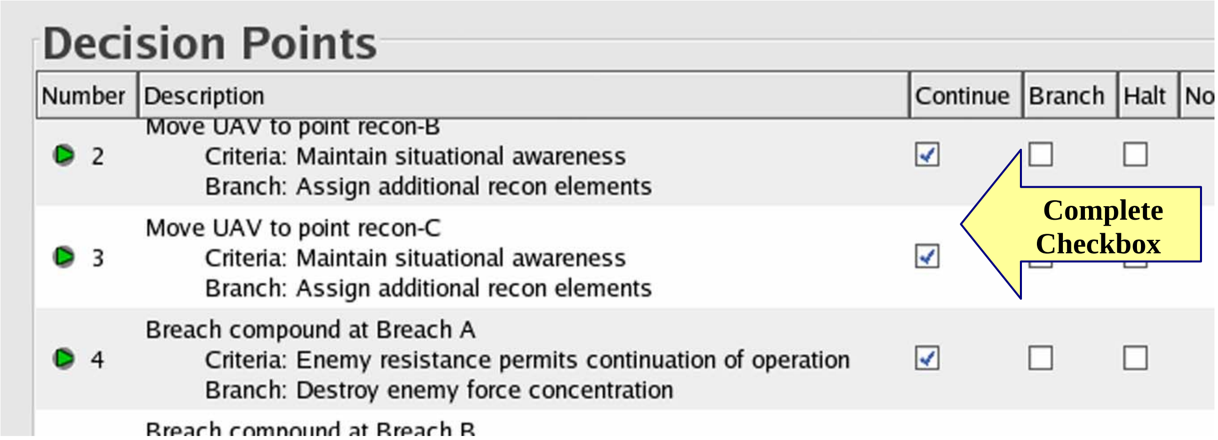


Figure A-8. Marking decision points as complete. The commander marks a decision point as complete by clicking the point's 'Continue' checkbox.

Managing Objectives

During plan execution, the commander has limited control of objectives. The commander's primary responsibility is to understand the objectives and how they relate to the plan, to monitor the points to know when key plan decisions must be made, to make appropriate plan decisions, and finally to make any appropriate adjustments that are required to successfully complete objectives. The list of objectives is preset in the plan. Currently the commander cannot add or remove objectives.

The CIANC^{3} system tracks the specific actions that are required to complete each objective, including the units assigned to the action, the tasks they are to complete, and their current status. The Objectives panel is setup as a table that provides the following information:

Table A-1

Objectives Panel Information

Activity:	Describes the primary effort being undertaken to accomplish the objective.
Action:	Describes the current effort being undertaken.
Units:	Identifies the friendly units assigned to support the objective.
Tasks:	Identifies the current tasking of the assigned units.
Decision Point:	Most objectives are tied to decision points. This field identifies the specific decision point to which an objective is tied.
Acceptable Loss:	The objectives that are associated with UAV's have an acceptable loss limit. This ratio places a limit on how many UAV's the CIANC^{3} system will task to an objective.

Changing the Acceptable Loss Limit

The commander can change the acceptable loss limit for an objective at any time before or after that limit is reached. To change the limit, the commander clicks on the [Edit] icon in the objective's acceptable loss field. The commander can then type in a new acceptable loss limit.

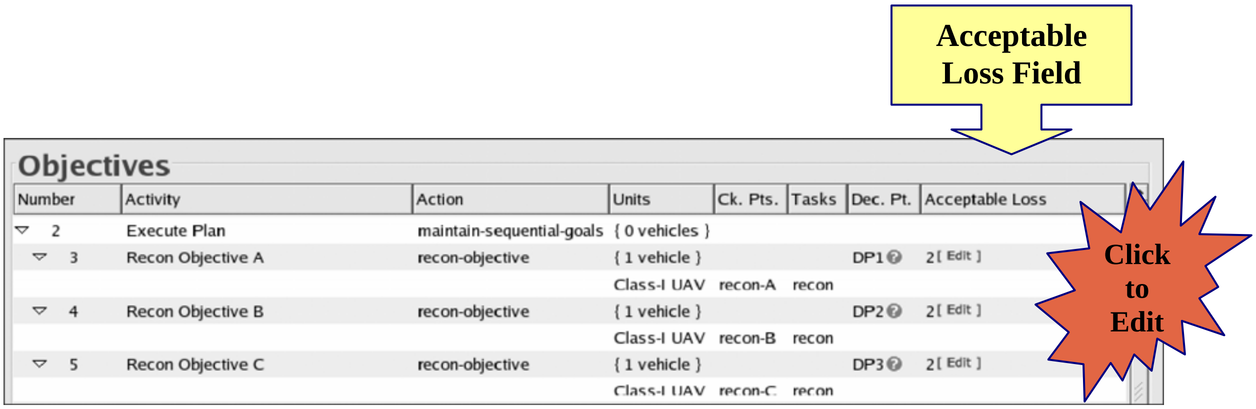


Figure A-9. Changing the acceptable loss limit. To change the limit, the commander clicks on the edit icon in the objective's acceptable loss field.

Objective Status

The CIANC[3] will mark objectives with the following status icons:

Waiting: Objectives begin in a waiting state and remain that way until the prerequisite requirements, including decision points and unit availability, have been met. The "waiting" status does not have an icon.

Executing: Objectives are "executing" when all prerequisite requirements, including decision points and unit availability, have been met and the system has successfully assigned available units to the objective tasks. The 'Executing' icon is a green arrow.

Completed: Objectives are "completed" when all tasks associated with the objective have been successfully accomplished. The 'Completed' icon is a black checkmark.

Failed/Aborted: Objectives have "failed" when the CIANC[3] system determines that a critical task or set of tasks cannot be accomplished. The 'Failed' icon is a red checkmark.

Managing Units Assignment

Available Units

A core part of mission execution that is not considered part of the plan is the mission unit assignments. This is because the CIANC[3] system has adopted a "just-in-time" approach to unit assignment based on the Department of Defense doctrine of Network Centric Warfare. The result is that CIANC[3] will always attempt to assign the most appropriate available units to a task at the time that the task needs to be completed. To do this the system maintains a listing of known friendly Units, their capabilities and dispositions. This listing is presented to the user in the Plan Display's Units panel.

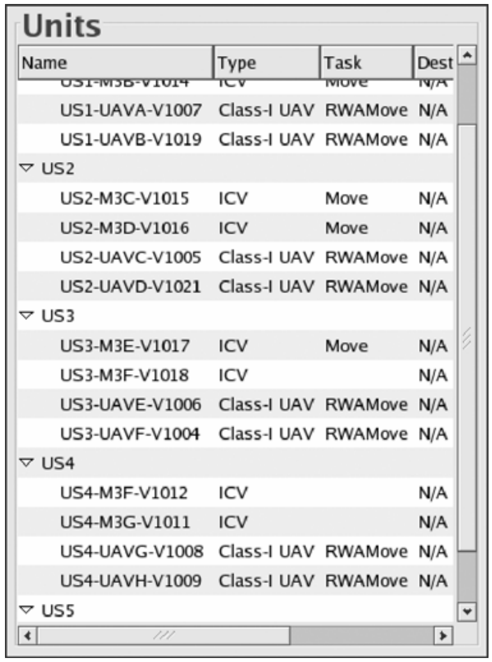


Figure A-10. Units Panel displays the status of known friendly assets.

Unit Assignment

The CIANC^3 system assigns units to tasks at three points in time.

1. CIANC^3 makes an initial assignment during the pre-operations phase. The commander triggers this assignment during the commander's evaluation of the operations plan. This initial assignment is to ensure that the tasks can be completed with existing units, and to provide the commander with an initial idea of how the plan will be executed.
2. When CIANC^3 is ready to make a task assignment, it rechecks the available units list to ensure that the best unit is being allocated to the task.
3. CIANC^3 monitors task units to ensure that they remain capable of completing their assignments. If at any time they become unable to complete their task, their task will be re-assigned to an appropriate unit.

The commander's role in unit assignment

The commander is involved in unit assignment in two ways.

Requesting Initial Unit Assignment: When the commander has reviewed and approved the plan, the commander requests initial unit tasking by clicking the 'Request Unit Tasking' button in the COP Display's Plan Control panel.

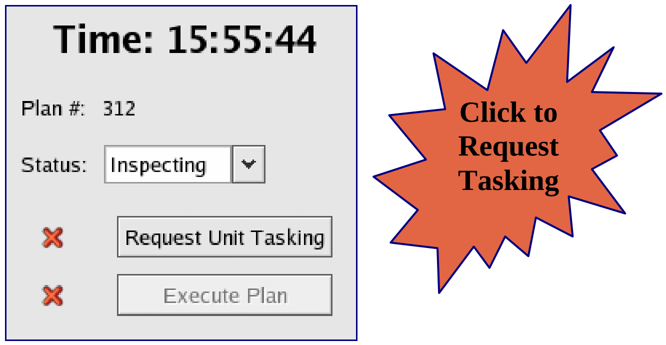


Figure A-11. Requesting initial unit assignments. When the commander has reviewed and approved the plan, the commander requests initial unit tasking by clicking the 'Request Unit Tasking' button in the COP Display's Plan Control panel.

Inspecting Initial Unit Assignments: The commander may inspect the initial unit tasking by looking at the Plan Display's Objectives Panel and Units Panel.

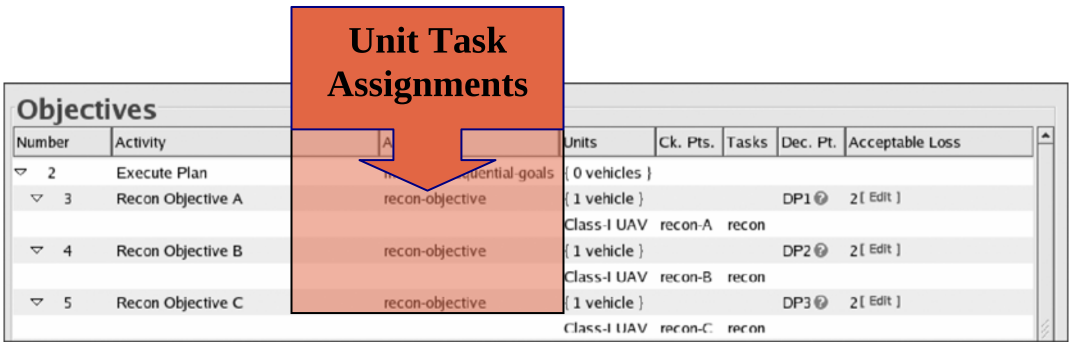


Figure A-12. Inspecting initial unit assignments. The commander may inspect the initial unit tasking by looking at the Plan Display's Objectives Panel and Units Panel.

Managing Routes and Checkpoints

All unit tasks are performed relative to pre-defined routes and checkpoints. Both routes and checkpoints are developed as part of the plan definition process. In the current evaluation software, initial routes and checkpoints are provided to the commander. During plan execution, the commander has the ability move most checkpoints to new locations.

Identifying Routes and Checkpoints

The current plan uses two kinds of routes: Approach Routes and Recon Routes. These are shown on the map using standard U.S. Army operation graphics. Approach Routes are drawn using a wide, open arrow. Recon Routes are drawn using a lightning bolt arrow. Checkpoints are drawn with black circles.

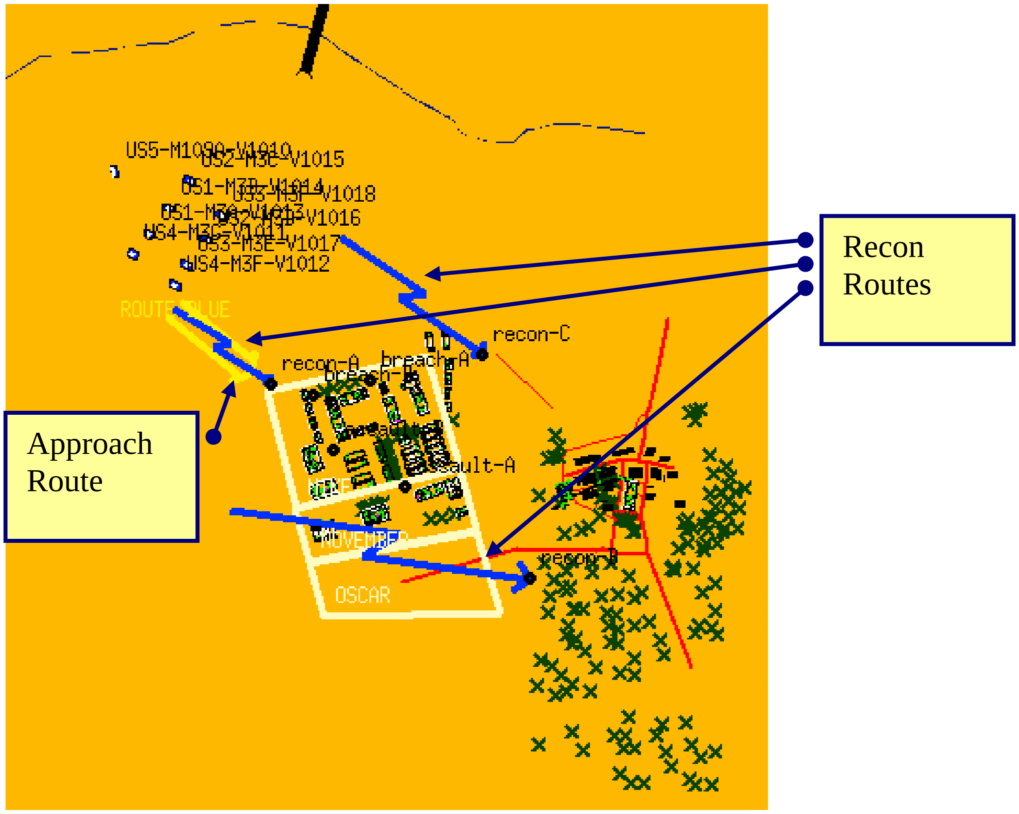


Figure A-13. Identifying plan routes. Approach Routes are drawn using a wide, open arrow. Recon Routes are drawn using a lightning bolt arrow.

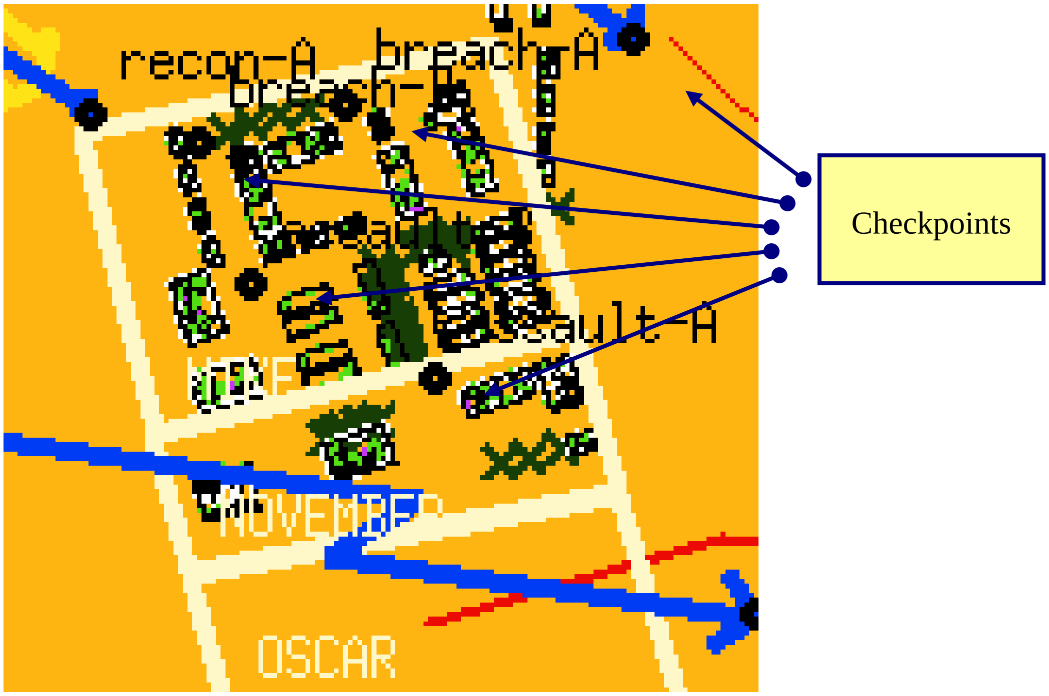


Figure A-14. Identifying checkpoints. Close-up of map showing Approach Routes and Recon Routes. Checkpoints are drawn with black circles.

Changing Routes and Checkpoints

In the current evaluation system, checkpoints may be moved, but not created or destroyed. Routes may not be created, deleted or directly modified. They will only change if associated checkpoints are relocated. Checkpoints may be moved in two ways: via the COP map and via the Checkpoints panel on the Plan Display.

Based on Visual Map Location: To change a checkpoint based on visual map location, the commander first locates the desired checkpoint on the map. Then the commander clicks on the checkpoint and drags it to the desired location.

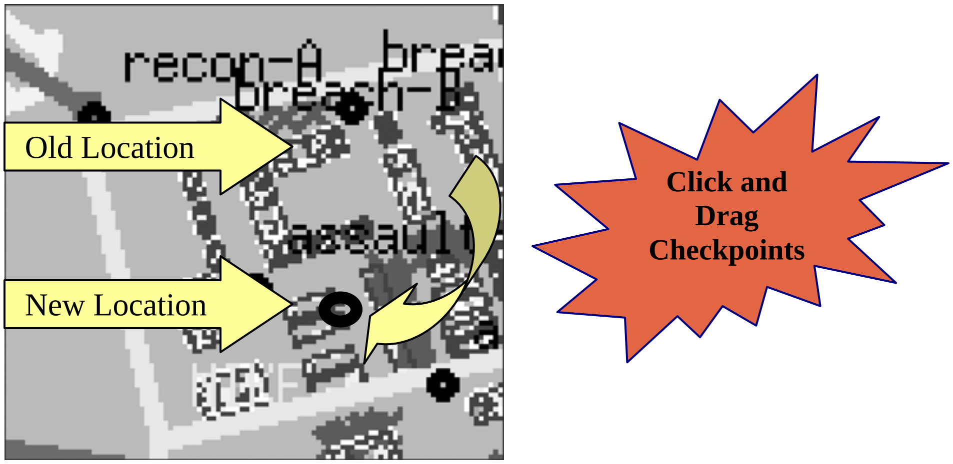
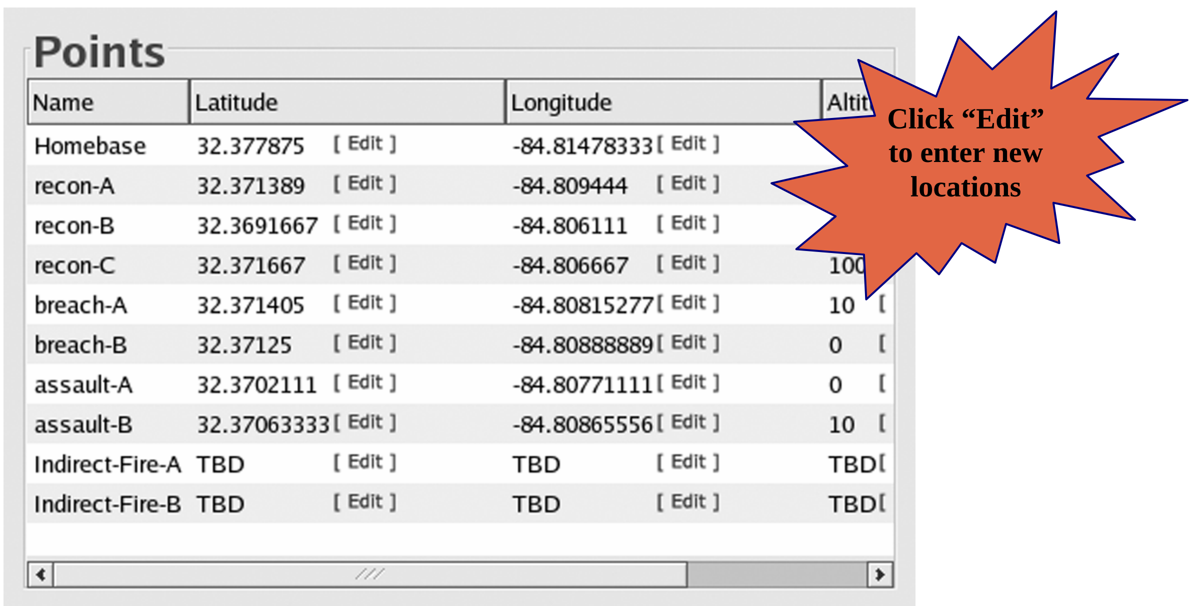


Figure A-15. Changing checkpoint location based on visual map location the commander clicks on the checkpoint and drags it to the desired location.

Based on Precise Map Coordinates: To change a checkpoint based on precise map coordinates, the commander first locates the desired checkpoint on the Checkpoints panel. The commander then clicks on the [Edit] icon in the desired latitude or longitude field. This will enable the field to be editable, allowing the entry of the desired coordinates. Changes to coordinates in the COP or Checkpoints panel are immediately relayed to all units whose tasking involves the checkpoint.



Points

Name	Latitude		Longitude		Altit
Homebase	32.377875	[Edit]	-84.81478333	[Edit]	
recon-A	32.371389	[Edit]	-84.809444	[Edit]	
recon-B	32.3691667	[Edit]	-84.806111	[Edit]	
recon-C	32.371667	[Edit]	-84.806667	[Edit]	100
breach-A	32.371405	[Edit]	-84.80815277	[Edit]	10 [
breach-B	32.37125	[Edit]	-84.80888889	[Edit]	0 [
assault-A	32.3702111	[Edit]	-84.80771111	[Edit]	0 [
assault-B	32.37063333	[Edit]	-84.80865556	[Edit]	10 [
Indirect-Fire-A	TBD	[Edit]	TBD	[Edit]	TBD [
Indirect-Fire-B	TBD	[Edit]	TBD	[Edit]	TBD [

Figure A-16. Checkpoints Panel. To change a checkpoint based on precise map coordinates, the commander clicks on the edit icon in the desired latitude or longitude field. Changes to coordinates in the COP or Checkpoints panel are immediately relayed to all units whose tasking involves the checkpoint.

The Common Operating Picture (COP) Display provides a map-based visualization of the battlespace. This visualization provides terrain information, built features such as buildings and roadways, Course of Action graphics including Route and Checkpoint markers, and friendly and enemy force locations. The COP Display also provides a message window that provides CCIR and other time-based notifications of plan status.

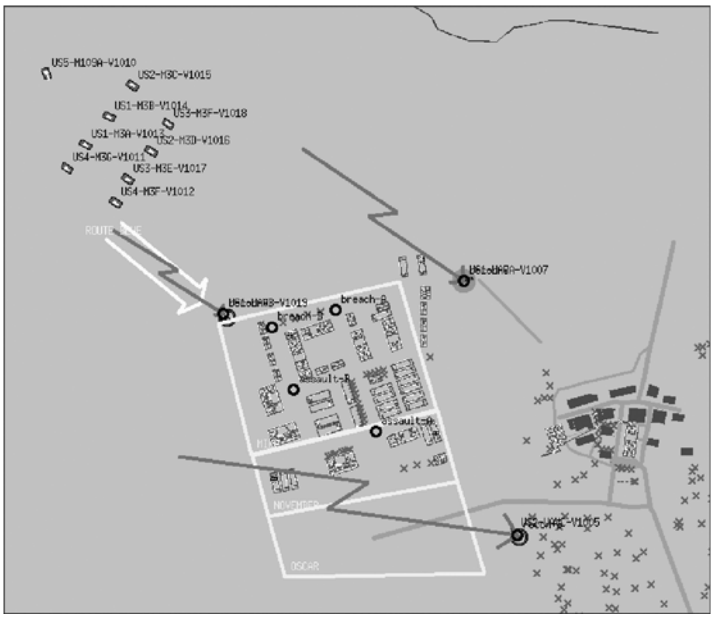
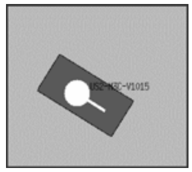
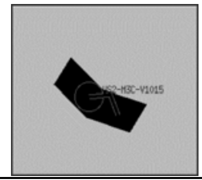
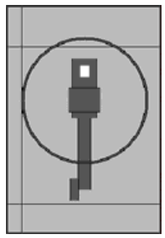
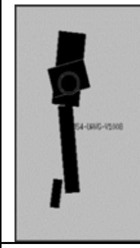
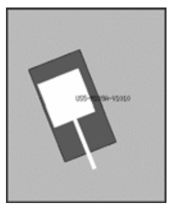
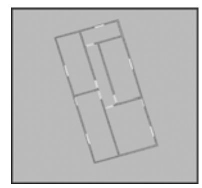
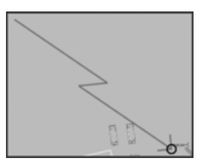
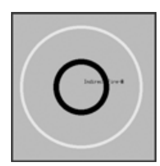
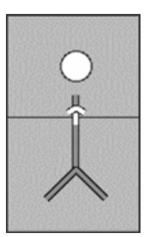


Figure A-17. Monitoring the COP. A commanders' view of the battlespace.

The COP is intended to present a legitimate commanders' view of the battlespace, incorporating information that the commander would have available to him or her in the course of battle. This includes friendly, but not enemy, fire indicators and friendly sensor areas. Currently, this does not include friendly weapon ranges.

Table A-2

Icons used in CIANC[3] COP within Map Display

ICV Troop Carrier		Destroyed ICV	
Class I Rotary-Wing UAV		Destroyed UAV	
Mounted Combat System		Building	
Fire Indicator (appears on both shooter and target)		Checkpoint Marker	
Recon Route Marker		Selection Indicator (selected checkpoint at left)	
Opposing Force Infantry			

Panning and Zooming

Panning is changing the area of map displayed in the COP without changing the resolution of the display. Zooming is changing the area of the map displayed by changing the resolution of display. The Pan and Zoom controls are in the upper right corner of the COP Display. The COP Map Display has one mechanism for panning and two for zooming.

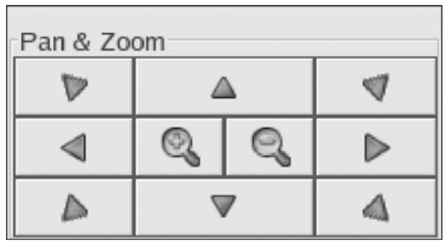


Figure A-18. COP Pan & Zoom controls.

Panning. To pan the COP Display, the commander will click on the Pan arrow that points in the desired direction. This will pan the display in that direction. More than one click may be necessary to move the display to the desired location.

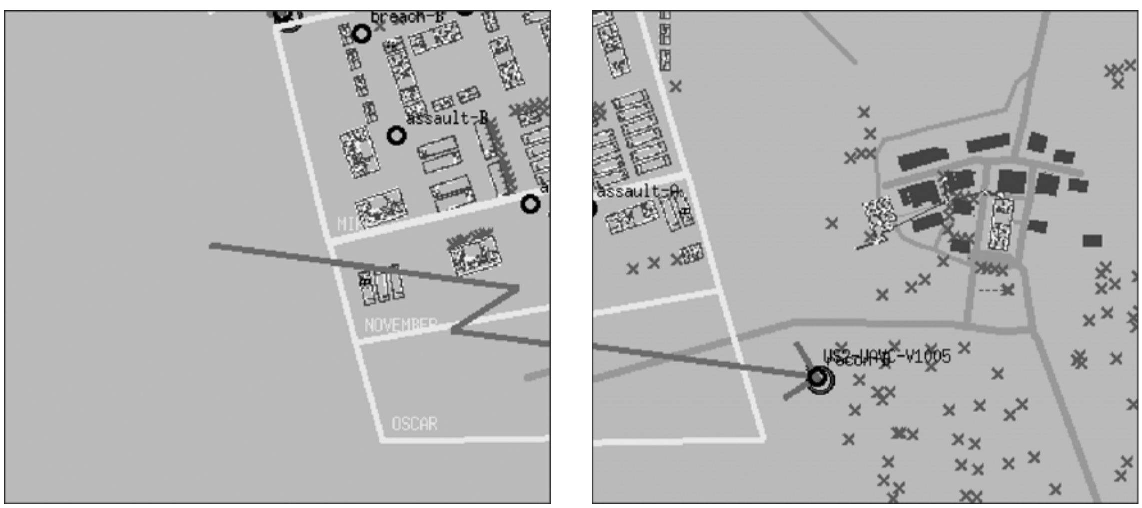


Figure A-19. COP Display before and after panning to the right.

Zooming. To zoom the COP Display, the commander will click on the Zoom magnifying glass. Clicking on the magnifying glass with the plus '+' sign, increases the resolution of image, decreasing the area observed. Clicking on the magnifying glass with the minus '-' sign, decreases the resolution of image, increasing the area observed. More than one click may be necessary to zoom the display to the desired resolution.

Figure A-20. COP Display before and after zooming.

Background Questionnaire

System Being Evaluated: Robotic Command and Control (CIANC3) by Soar Technology
Session: ___________ **Participant ID:** ________

Demographics

1. Age Range *(please check one)*
   Below 22 22 - 25 26 - 29 30 – 34 over 34
2. Gender *(please check one)*
   male female
3. Officer Grade (i.e., O-1, O-3): ________________________
4. Unit: ____________________________________

5. What is your Military Occupational Specialty? ________________________________

6. What schools have you completed or are currently attending?

a.	e
b.	f
c.	g
d.	h.

7. Military History:

- ____________________________________
- ____________________________________
- ____________________________________
- ____________________________________
- ____________________________________
- ____________________________________
- ____________________________________
- ____________________________________
- ____________________________________
- ____________________________________
- ____________________________________

Example

- At 18, enlisted in Army (1992) MOS Artillery
- Assigned to 94 ID.
- Honorably discharged in 1996 at E-3
- Enrolled in Georgia Tech Army ROTC
- Graduated in 2000, commissioned as O-1
- Trained in 2nd ACR, O-2 in 2002.
- O-3 in 2004 and assigned as HHD CO for 123 Finance Battalion, 3 ID.

Computer Experience

8. Do you use a desktop computer in your work?
   Never   Occasionally   Weekly   Daily

9. Do you use a laptop computer in your work?
   Never   Occasionally   Weekly   Daily

10. Do you use military computer hardware (e.g. targeting computers) in your work?
    Never   Occasionally   Weekly   Daily

11. Do you use a computer outside of your work?
    Never   Occasionally   Weekly   Daily

12. Do you use computer games for <u>training</u>?
    Never   Occasionally   Weekly   Daily

13. What <u>training</u> games have you played recently:

a.	e
b.	f
c.	g

14. Do you play computer games for <u>entertainment</u>?
    Never   Occasionally   Weekly   Daily

15. What genre games do you play? (*Check all that apply)*
    Strategy   Role-Play Games   First Person Shooters
    Sports   Puzzle   Massive Multiplayer Online

16. What <u>entertainment</u> games have you played recently?

a.	e
b.	f
c.	g

17. What is your skill with computers?
    Novice   Low skill   Medium Skill   High Skill

Experience with Military Simulations

18. Have you participated in a computer-based military simulation?
    Yes No

19. Have you operated a computer in a computer-based military simulation?
    Yes No

20. If you answered Yes to question 19 above, what computer simulations and/or simulators have you operated?

a.	d.
b.	e.
c.	f.

Experience with Unmanned or Automated Systems

21. What is your level of experience with unmanned/teleoperated ground vehicles?
    None Low Medium Expert

If you have prior experience, please describe below:

22. What is your level of experience with unmanned air vehicles and sensors?
    None Low Medium Expert

If you have prior experience, please describe below:

SAGAT Questions

Version 0.1
Soar Technology

System Being Evaluated: CIANC^3
Session :
Participant ID :

SAGAT 1 Questions

1. Indicate the location(s) of each element on the map?

2. Which of the following assets are available to support you?
   a. NLOS Weapons
   b. Smoke
   c. Reinforcements
   d. UAV sensors
   e. None

3. Where are the principal enemy concentrations?

4. What do you expect the enemy to do in the next 5 minutes?
   a. Attack
   b. Nothing
   c. Move Positions
   d. Defend
   e. Retreat
   f. Other: ___________________________________

SAGAT Questions

Version 0.1
Soar Technology

System Being Evaluated: $CIANC^3$
Session :
Participant ID :

SAGAT 2 Questions

1. Which friendly forces are currently exposed to enemy fire?

2. Which enemy element is your highest-level threat?

3. How many casualties have you suffered? What level of asset loss?

4. On the map, indicate which enemy threats are currently under reconnaissance. Indicate those that are not.

Post Evaluation Questionnaire

Version 0.1
Soar Technology

System Being Evaluated: CIANC[3]
Session :
Participant ID :

System Behavior		**1**	**2**	**3**	**4**	**5**	**6**	**7**		**NA**
1. System behavior was understandable	Never								Always	
2. System behavior was predictable	Never								Always	
3. System behavior was controllable	Never								Always	
4. System behavior was appropriate	Never								Always	
5. Other Comments:										

System Concepts & Terminology		**1**	**2**	**3**	**4**	**5**	**6**	**7**		**NA**
6. System used familiar concepts	Poorly								Well	
7. Extensions to familiar concepts were	Confusing								Clear	
8. System used familiar terminology	Poorly								Well	
9. Extensions to familiar terminology were	Confusing								Clear	
10. System used familiar work procedures	Poorly								Well	
11. Extensions to familiar work procedures were	Confusing								Clear	
12. System organization supported tasks	Poorly								Well	
13. Other Comments:										

Information Presentation		**1**	**2**	**3**	**4**	**5**	**6**	**7**		**NA**
14. Information display was	Confusing								Clear	
15. Information display was	Insufficient								Sufficient	
16. Information display was	Irrelevant								Relevant	
17. Information display was	Frustrating								Satisfying	
18. Other Comments:										

Ease of Learning		1	2	3	4	5	6	7		NA
19. Learning to operate the system was	Difficult								Easy	
20. Remembering appropriate commands or controls was	Difficult								Easy	
21. Locating functions and information was	Difficult								Easy	
22. System messages helped learning	Minimally								Greatly	
23. Reference and training materials were	Confusing								Clear	
24. Training time was	Insufficient								Sufficient	
25. Other Comments:										

System Performance		1	2	3	4	5	6	7		NA
26. System speed was	Slow								Fast	
27. System reliability	Unreliable								Reliable	
28. Other Comments:										

General Reaction		1	2	3	4	5	6	7		NA
29. Overall reaction was	Negative								Positive	
30. Using the system was	Difficult								Easy	
31. Using the system was	Frustrating								Satisfying	
32. Using the system was	Boring								Engaging	
33. System was	Limited								Powerful	
34. System was	Rigid								Flexible	
35. System was	Inappropriate								Appropriate	
36. System was	Confusing								Clear	
37. Other Comments:										

38. What features seemed the most useful?

39. What features seemed the least useful?

40. What tasks, if any, were unacceptably difficult?

41. What tasks, if any, were unexpectedly easy?

42. What functionality or information should be added to the system to increase its' usefulness?

Group Discussion Survey

Participant:

1. What are the hardest situation awareness requirements for company-command?

    a. What information could the UAV or platoons have provided that would have improved SA?

    b. What information could have been provided about the UAV or platoon behavior that would have improved SA?

    c. What do you need to attain and maintain good SA?

    d. How do you visualize the tactical situation? How would you like to do this?

2. What types of decisions would you like help with?

3. What kind of things would you or would you not want the system to automate/make suggestions for? (on scale of 1 to 5 where 1 is low priority and 5 is high)
    a. Check point placement?
    b. Target pairing?
    c. Unit assignment?
    d. CCIRs & reports to Higher?
    e. Fire requests?
    f. Logistics & Re-supply?
    g. Movement/Hazard avoidance?

    h. What other high priority tasks would you add?

4. Thinking about using this tool in an operational environment (e.g., FBCB2) what issues might prevent a tool like this from being useful adopted?
    a. Comms overload?
    b. Mission Tempo?
    c. Mismatch with commanders responsibilities?
    d. Anything else?

5. We organized around objectives and decision points. Is that how you would do it?

    a. If not what would you suggest?

6. What tools would you like to see developed to help you?
    a. Train
    b. Plan
    c. Execute missions
    d. Conduct AARs.

Appendix F

CIANC^3 Software Introduction Script

Soar Technology
Version 0.1

Have Water Available
Have Manual Handy
Review Think Aloud Protocol
Review "Evaluating the System, not the User"

System Overview

The CIANC^3 system is designed to assist a U.S. Army company commander controlling multiple robotic entities. The current version of the software is designed to run in simulation, allowing the commander to direct simulated entities in a pre-defined mission. While controlling this mission, the commander is able to monitor relevant objectives and decision points, monitor the Common Operating Picture (COP) for current entity status, and make limited updates to the current plan. The system supports this process by dynamically allocating units as they are needed, monitoring status and resolving issues that do not affect the plan, and alerting the commander to plan progress.

We'll now go through the system displays one at a time, and I'll show you kinds of information and actions that the displays make available. You'll also be provided a reference manual that can be referred to at any time. Please feel free to ask questions at any time. I will answer your question if I can or request that you wait until we reach the part of this script that answers your question.

Main Display Areas

COP Display

This is the COP display. The COP display provides the main Map, Message, and Plan Control panels. These panels provide the primary orientation to the current battlespace as a mission is executed. The COP provides a standard map based view of the battlespace, showing the positions and dispositions of friendly and enemy forces, checkpoints, routes and Named Areas of Interest (NAI), as well as terrain features such as buildings, roadways and vegetation.

The Message Panel shows system-generated alerts, including unit task progress, commander critical information requirement (CCIR) notification, and decision point notifications. The Plan Control panel enables pre-operation management and control of the operation plan. I'll come back to this later.

Plan Display

This is the Plan Display. The Plan display provides detailed access to the current plan Decision Points and Objectives, as well as plan checkpoints and unit status. The Decision Points panel lists the current plan Decision Points and their status, allowing the commander to observe and control operation advancement. The Objectives panel shows specific named objectives and tasked units and their current actions toward the objective. The Checkpoints panel provides a listing of checkpoints identified in the current plan and their locations. The Units panel provides additional detail about simulated blue force units and their location and current disposition.

CCIRs

Unlike decision points and objectives, the CIANC^3 Plan Display does not display the current CCIR list. Instead, the commander is notified about CCIR status changes via the COP display message. CCIRS and other system messages are currently color coded by level of severity, red being the highest and blue the lowest.

Executing the plan

To execute a mission, the commander must first review the current plan and familiarize himself with its contents. This plan will include general descriptions of the types of units required to complete each objective. At the direction of the commander, the system identifies and tasks available units in manner consistent with the plan.

Requesting Initial Unit Assignment: The commander requests initial unit tasking by clicking the 'Request Unit Tasking' button in the COP Display's Plan Control panel.

Once the commander has reviewed and is satisfied with the plan and unit assignments, the commander may execute the plan. Executing the plan triggers the system to direct assigned units to complete the first set of objectives and to start monitoring the first decision points.

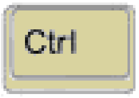
Executing Plan: To execute the plan, the commander presses the 'Execute Plan' button located on the upper right corner of the COP Display. Once the button is pressed the system will immediately direct units to act.

Managing Decision Points

During plan execution, the commander has limited control of decisions points. The list of decisions points is preset in the plan. Currently the commander cannot add or remove decisions points. The commander's primary responsibility is to understand the decision points and how they relate to the plan, to monitor the points to know when key plan decisions must be made, to make appropriate plan decisions, and finally to mark the decision points with the results of those decisions.

CIANC^3 system progresses through the plan at a pace determined by the commander. The commander exercises control over plan execution by managing decision points. By marking decision points as being successfully completed, the commander authorizes future mission progress. Pending this approval, the CIANC^3 system will not have units initiate new action.

Decision Point Notification: CIANC^3 will notify the commander when the system believes that a decision point has been reached, by highlighting the decision point and displaying "pending" in the Decision Point panel's 'Notes' field.

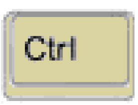


Marking Decision Points as Complete: The commander marks a decision point as complete by clicking the points' "Continue" checkbox, as in the example below.

Managing Objectives

During plan execution, the commander has limited control of objectives. The commander's primary responsibility is to understand the objectives and how they relate to the plan, to monitor the points to know when key plan decisions must be made, to make appropriate plan decisions, and finally to make any appropriate adjustments that are required to successfully complete objectives. The list of objectives is preset in the plan. Currently the commander cannot add or remove objectives.

The CIANC^3 system tracks the specific actions that are required to complete each objective, including the units assigned to the action, the tasks they are to complete, and their current status. The objectives that are associated with UAV's have an acceptable loss limit. This ratio places a limit on how many UAV's the CIANC^3 system will task to an objective. This metric is tied to the UAV Loss CCIR and will warn the commander when the loss limit has been reached.



Changing the Acceptable Loss Limit: The commander can change the acceptable loss limit for an objective at any time before or after that limit is reached. To change the limit, the commander clicks on the [Edit] icon in the objective's acceptable loss field. The commander can then type in a new acceptable loss limit.

Managing Routes and Checkpoints

All unit tasks are performed relative to pre-defined routes and checkpoints. Both routes and checkpoints are developed as part of the plan definition process. In the current evaluation software, initial routes and checkpoints are provided to the commander. During plan execution, the commander has the ability move most checkpoints to new locations.

Identifying Routes and Checkpoints

The current plan uses two kinds of routes; Approach Routes and Recon Routes. These are shown on the map using standard U.S. Army operation graphics. Approach Routes are drawn using a wide, open arrow. Recon Routes are drawn using a lightning bolt arrow. Checkpoints are drawn with black circles.

Changing Routes and Checkpoints

In the current evaluation system, checkpoints may be moved, but not created or destroyed. Routes may not be created, deleted or directly modified. They will only change if associated checkpoints are relocated. Checkpoints may be moved in two ways. First, via the COP map, and second, via the Checkpoints panel on the Plan display.

Changing Checkpoint Location Based on Visual Map Location: To change a checkpoint based on visual map location, the commander first locates the desired checkpoint on the map. Then the commander clicks on the checkpoint and drags it to the desired location.

The COP is intended to present a legitimate commander's view of the battlespace, incorporating information that the commander would have available to him or her in the course of battle. This includes friendly, but not enemy, fire indicators and friendly sensor areas. Currently, this does not include friendly weapon ranges.

Panning and Zooming

Panning is changing the area of map displayed in the COP without changing the resolution of the display. Zooming is changing the area of the map displayed by changing the resolution of display.

The Pan and Zoom controls in the upper right corner of the COP display control panning. To pan the COP display, the commander will click on the Pan arrow that points in the desired direction. This will pan the display in that direction. More than one click may be necessary to move the display to the desired location.

To zoom the COP display, the commander will click on the Zoom magnifying glass. Clicking on the magnifying glass with the plus '+' sign, increases the level of detail and decreases the area observed. Clicking on the magnifying glass with the minus '-' sign, decreases the level of detail of image and increases the area observed. More than one click may be necessary to zoom the display to the desired resolution.

Review Iconography
Review Sensor Range Rings

Example Tasks:

Now that we've gone through the basics of the system, we will go through 4 short examples to help you familiarize yourself with the display.

1 According to the Decision Points Panel how many Recon Points will be covered by UAVs?
2 According to the Objectives Panel, what is the Acceptable Loss level for each UAV?

Request Initial Tasking

3 According to the Objectives Panel, which UAV's are assigned to which Recon Point?

4 According to the COP, where are these recon points?

Appendix G

Mission Details

Soar Technology

CCIRs

CCIR 1	(PIR)	Report any movement of enemy or unidentified elements
CCIR 2	(PIR)	Report any obstacles or ambushes
CCIR 3	(PIR)	Report any use of chemical agents
CCIR 4	(PIR)	Report successful breach of compound
CCIR 5	(PIR)	Report any enemy contact
CCIR 6	(PIR)	Report successful entry of communications center
CCIR 7	(PIR)	Report enemy snipers
CCIR 8	(PIR)	Report enemy armor
CCIR 9	(FFIR)	Report any friendly unit or lag status below 50%
CCIR 10	(FFIR)	Report loss of any UAV

Objectives

#	Activity	Unit	CKPT	DCPT
1	Recon Objective A	Class-1 UAV	recon-A	DP1
2	Recon Objective B	Class-1 UAV	recon-B	DP2
3	Recon Objective C	Class-1 UAV	recon-C	DP3
4	Call Indirect Fire		Indirect-Fire-A, Indirect-Fire-B	DP4
5	Breach Objective A	ICV	Breach-A	DP5
6	Breach Objective B	ICV	Breach-B	DP6
7	Assault: Move assaulters into position	ICV	Assault-A	DP7
8	Assault: Move assaulters into position	ICV	Assault-B	DP8
9	Assault: Assault Point A	ICV	Assault-A	DP9
10	Assault: Assault Point B	ICV	Assault-B	DP10

Decision Points

#	**Description**	**Criteria**
1	Move UAV to point recon-A	Maintain situational awareness
2	Move UAV to point recon-B	Maintain situational awareness
3	Move UAV to point recon-C	Maintain situational awareness
4	Indirect fire at Indirect-Fire-A and Indirect-Fire-B	Enemy resistance permits continuation of operation
5	Breach compound at Breach-A	Enemy resistance permits continuation of operation
6	Breach compound at Breach-B	Enemy resistance permits continuation of operation
7	Reinforce Breach A with Assault Team	Enemy resistance permits continuation of operation
8	Reinforce Breach B with Assault Team	Enemy resistance permits continuation of operation
9	Assault compound at Assault-A	Enemy resistance permits continuation of operation
10	Assault compound at Assault-B	Enemy resistance permits continuation of operation

Appendix H

Observation & Verbal Protocol Analysis

Observation and Verbal Protocol Analysis is that portion of a Task Analysis based upon close observation of users doing specified tasks while encouraged to "think aloud" during the performance of those tasks. These Observations are noted, subjected to Interpretation, and Recommendations are then generated.

Table H-1

Automation Observations and Recommendations

Observation	Interpretation	Recommendation
Most participants expressed the view that the system allowed insufficient control of entities. This includes unit task assignment, unit movement, and unit positioning. It also includes allowing the participant to alter the plans and decision points while the plan is executing.	Automation may or may not be useful for entity control, but the commander must feel that he or she has the ultimate control over their assets.	Ensure that automation suggestions can be overridden and that the command can make all decisions manually if desired. Ensure that automation plans allow fine adjustments, not just full acceptance/rejection.
Many participants expressed the view that automation support was potentially useful, particularly if it provided doctrinally correct suggestions and could explain rationale.	Automation may be useful if it reliably (trust) provides correct (competent) results.	Ensure when automation is included in a system, it must be sufficiently smart to provide acceptable answers and explanations. This will require including knowledge of doctrine and situation into decision process.
Most participants made heavy use of recon point location to guide UAV behavior, using direct manipulation to move the recon point around the Map Display. The participants that used this technique appreciated it greatly.	This was a successful and somewhat surprising approach to automation control invented by the participants. It was surprising because the plan, as defined in the OPORD and system, only required the points be moved to appropriate reconnaissance positions once. We did not anticipate the participants using this as a means of direct control. Using the system in this way points to two things: 1) There should have been more opportunity for UAV control built into the plan and supported by the system. 2) This direct manipulation style was very effective, limited by the system's inability to provide live sensor displays from the UAV's point of view.	Ensure that the user has multiple mechanisms for intervening in system behavior, from planning to execution phases, and from formal plan-based control, to grab-the-stick control.
Many participants suggested or responded favorably to extensions of the automation included in evaluation system; including route planning, identification of high-value targets, logistics, weapon pairing, and CCIR management.	Participants, with caveats listed above, are very interested in evaluating and acquiring support tools.	Select forms of automation other than asset selection to be incorporated into future versions.

Table H-2

GUI Design Observations and Recommendations

Observation	Interpretation	Recommendation
Most participants were able to make use of the Map and Plan Displays quickly, with minimal training.	Schema-centric/decision-centric design approach provided solid linking of participant and agents, with few requirements to learn new concepts.	Continue to draw on schema-centric design approach relative to agent-based interface design.
Participants had mixed success with and enthusiasm for CCIR/system message display. Complaints tended to focus on difficulty of managing the interface and associating with other displays. Automatically identifying and displaying CCIRs (Commander's Critical Information Requirements) and other IR types was well received.	The concepts expressed by the message console are potentially useful, but the specific GUI design implemented was visually confusing, not salient enough, hard to operate, and at times redundant or disjoint from other data provided by the system.	Improve message delivery concept by better integrating with other displays and improving visual coding, salience and organization.
With the exception of the CheckPoints pane, each of the Plan Display panes was used regularly.	The lack of use of the CheckPoints display points to a possible difference in GUI requirements between mission planning, where such a display was useful (to the system designers), and mission execution, where it wasn't useful to the participants.	Move CheckPoints pane off the main Plan Display to a secondary screen available by request. This screen might have other panes relevant to mission planning.
Participants made heavy use of both Map Display and Plan Display. Participants made a number of requests for better integration of the Plan Display and the Map Display. Specific requests tended to focus on data missing from the Map Display, such as Decision Points.	There is a set of data with obvious geographic anchors (such as Decision Points) that should available to the participant when he is thinking visually or geospatially. There are also circumstances when thinking about the mission non-geospatially is important. For example, scanning a list of assets for the status is much more efficient that scanning a map for the same complementary forms information.	Ensure that all data with obvious geospatial anchors is viewable on the Map Display. Allow layer control to show/hide this data. Ideally, data grouping by layer should be based on a task analysis. Ensure that primary data sets are available in non-geospatial forms to support other modes of situation assessment. In next evaluation round, establish whether better map-based display decreases usage of Plan Display. This idea was suggested by some observers but rejected by others.

Table H-3

Information Design Observations and Recommendations

Observation	Interpretation	Recommendation
Most participants had specific issues and/or made specific complaints about information that was missing from the Map Display and Plan Display. Specific examples include: terrain information and contour lines; building elevations; gun tube orientation; enemy and friendly line-of-sight; dismount and UGV locations; unit capabilities, damage, ammunition, and fuel status.	We expected that our current information display would only be marginally sufficient for the evaluation exercised, and we were right. Most of the additional data requested are available to the system and could be displayed by the system.	Based on task analysis, ensure that all mission critical information requirements met. Use layering and other techniques to manage visual clutter as necessary.
Almost half of the participants had specific issues and/or made specific complaints about the lack of raw sensor information. While they appreciated the integrated COP, they felt that they needed the opportunity to inspect the raw sensor reports. This came up in a number of places, including surveillance/reconnaissance and Battle Damage Assessment.	Participants had two issues. First, they did not trust the information viewable on the COP as being truly accurate and wanted to be able to confirm it. Second, the level of detail provided on the COP was not always sufficient (or did not appear to be sufficient) for their current task.	Based on task analysis, ensure that all mission critical information requirements are met. Add raw, or partially processed, sensor-specific displays where possible and appropriate. Add sensor coverage & pedigree capabilities (not pedigree information) where possible and appropriate
Most participants requested additional information about enemy capabilities and behavior.	There are a number of issues at play here. The evaluation system should provide as much information as would available in a deployed system but not more. The use of virtual sensors with better capabilities than exist in real systems may please the participants but is not helpful to the evaluation (unless evaluating the impact of the new sensors is part of the evaluation) The evaluation system should also not provide less information. One critical information source not adequately provided in this simulation is SPOT reports from human Soldiers. The participants seemed to suffer at points by not receiving information that they would have expected.	There is a limit to the fidelity of the simulation, but to the degree possible the system should provide as accurate representation of sensor data as possible. Ensure that the participants receive information in the format and quality they expect, including SPOT reports.
The participants had mixed reaction to the icons and visual coding schemes used in the system. For example, the icons used to represent friendly and enemy units were non-standard and ambiguous, and platoons were not given standard call-signs and color codes.	The icons used in the system were a combination of standard military operational graphics (FM-105-5-1) and graphics supplied by the simulation system (OneSAF Test-Bed). In many cases the OTB graphic symbols represented new data types that have not been standardized within the U.S. Army.	Ensure that the system uses military standard graphics when possible; FM-105-5-1 and MILSPEC 2525b in particular. Ensure that there are secondary interaction mechanisms within the system to identify the meaning of a graphic. For example, allowing a mouse-over event to trigger a hovering screen message. Ensure that adequate training time

Table Continues

Observation	Interpretation	Recommendation
		and reference materials are provided for participants to familiarize themselves with the graphics.
At least two participants noted that route information was not useful during mission execution.	Like other information graphics, route graphics are useful in some instances and not others.	Use layering and other techniques to manage visual clutter as necessary.
Most of the participants had specific issues and/or made specific complaints with the Plan Display symbology and color-coding.	The Plan Display symbology was focused on making key information salient. While it was generally successful at that, in some places it failed at clarifying what the salient information meant.	Make sure that alerts and other salience-encodings point to displays that clearly articulate their meaning.

Table H-4

Miscellaneous Observations and Recommendations

Observation	Interpretation	Recommendation
Most participants spent a considerable amount of time verbally reporting CCIRs to higher echelons, and in some cases, commenting on how they would like to be communicating with other companies either for relief or flanking coverage.	Communication is a major effort that requires substantially more support than provided in the current system. Communication demands may also place limits on usefulness of this type of system.	Conduct follow-up evaluation where participant has explicit communication requirements and load placed on him. Analyze system for opportunities to support & simplify communications. Will shared COP reduce need for communication? (no) Can system automate CCIR reporting process?

Focus Group Questionnaire

The final session conducted consisted of a focus group made up of two captains and two second lieutenants. The following table describes their responses to a focus group only questionnaire.

Table H-5

Focus Group Observations and Recommendations

Observation	Interpretation	Recommendation
Participants all described location of friendly and enemy forces as the hardest company-command SA requirement.	Based on these comments and others made during the evaluations, it is clear that understanding the physical location of troops and enemy forces is critical for decisions ranging from operation pacing and decision points, to weapon-target pairing and logistics.	Ensure that any display provided the commander show, to the level of detail available, all location data. Ensure that any display provided the commander directly shows the commanders' geo-spatial decision-making. For example, calculate and display weapon-target pairing blast circles around identified target points, and/or automatically select weapons to match target and blue force locations.
Participants commented on the need for raw or semi-processed UAV sensor output, not just the fully integrated COP.	There is a combination of two factors underlying this request. First, there is a general lack of trust for the COP Display. For any important decision, the commanders wanted to examine the pedigree of the information provided. Also, the COP, as a top-down map view, only provides a small, symbolized portion of the data available from the UAV sensors. The commanders wanted to use the UAVs for BDA and to help orient them to the situation on the ground in ways they did not feel the map supported.	Supplement the COP map display with secondary display(s) that show the camera view of the UAV. Even if this is only in text (based on simulation output) it would be an improvement.
Participants could not monitor UAV status, capabilities and behavior.	The UAV's were difficult to manage because the commanders did not have a view of any UAV properties other than physical location and sensor radius. They requested all of the expected information, including fuel, payload and default reaction behavior (Run, Evade, Engage, Designate).	Along with the UAV Sensor Display discussed above, display more information and provide more control over the individual UAVs.

Table Continues

Observation	Interpretation	Recommendation
When asked, in the questionnaire and verbally, what kind of decisions the commanders would like help with, the response was generally "all of them." Specific things that came up include: Route feasibility/Hazard avoidance Refueling point planning, logistics planning, maintenance planning CCIRs and reports to higher echelons Target pairing & Fire requests	The commanders acknowledge that they had a hard job, under hard conditions, with fatal consequences for bad decisions. As long as it was managed well and under their control (i.e., if it was competent and they trusted it), they'd use any support that was made available. They would be happy with a tool that could quickly give them doctrinally correct answers that they could accept, reject or improve.	Ensure that automation/decision support produces reliable doctrinally correct suggestions before it is field tested or fielded. There should be no limits on the automation research. The commanders placed no restriction on aspects of their job that could be supported.
When asked, in the questionnaire and verbally, what characteristics of an operational environment might prevent usage of a tool like this, the two primary responses were communications overload and mission tempo.	In current operations, the commanders spend a substantial portion of their time communicating via radio with higher and lower echelons, as well as with other peer echelons. This communication is critical to maintain organizational SA and to deliver operational orders. Their concern is that they spend so much time doing this coordination that they might not be able to manage this system as well and maintain mission tempo. This may not be a future concern, though. The DoD goal is that much of the current analog radio chatter should be replaced by system-to-system digital communications, lowering the communications burden on the commander.	Evaluate this system and systems of its type in high and low communication requirement exercises to gauge whether this is a real concern. Ensure that this system integrates with a Global Information Grid/Distributed Digital COP. Where possible, ensure that the system supports automation of CCIR reporting and other reports to higher echelons.

Appendix I

Acronym List

ACL	Agent Communication Language
AOC	Area of Concentration
AEW	Airborne Early Warning
ARI	U.S. Army Research Institute for the Behavioral and Social Sciences
ARL	Army Research Laboratory
BBS	Brigade/Battalion Simulation
BINAH	Battlespace Information and Notification through Adaptive Heuristics
BLOS	Beyond Line of Sight
C3	Command, Control, and Communications
C4ISR	Command, Control, Communications, Computers, Intelligence, Surveillance, Reconnaissance
CCIR	Commander's Critical Information Requirement
CERDEC	U.S. Army Communications - Electronics Research, Development, and Engineering Center
CIANC[3]	Cooperative Interface Agents for Networked Command, Control and Communications
COA	Course of Action
CoABS	Control of Agent Based Systems
COP	Common Operational Picture
CCTT	Close Combat Tactical Trainer
DAML+OIL	DARPA Agent Markup Language plus Ontology Inference Language
DCA	Defensive-Counter Air
DOD	Department of Defense
FCS	Future Combat Systems
FIPA	Foundation for Intelligent Physical Agent
FWA	Fixed Wing Aircraft
GIG	Global Information Grid
GUI	Graphical User Interface
HLA	High Level Architecture
HLSR	High-Level Symbolic Representation
ICF	Intelligent Control Framework
IFF	Identification, Friend or Foe
IR	Infrared

JI	Joint Intention
JSAF	Joint Semi-Automated Forces
JV2020	Joint Vision 2020
KEUA	Knowledge Enablers for Unit of Action
MDMP	Military Decision-Making Process
MVC	Model-View-Controller
NAI	Named Areas of Interest
NCW	Network-Centric Warfare
OneSAF	One Semi-Automated Force
ONR	Office of Naval Research
OPORD	Operation/Operational Order
OSD	Office of Secretary of Defense
OTB 1.0	OneSAF Testbed Baseline
ROCCIE	Robotic Command and Control Intelligent Enablers
ROE	Rules of Engagement
RWA	Rotary Wing Aircraft
SA	Situation Awareness
SAGAT	Situation Awareness Global Assessment Technique
SBIR	Small Business Innovation Research
STX	Situational Training Exercise
TACOPS	Tactical Operations
TARDEC	U.S. Army Tank Automotive Research, Development, and Engineering Center
TCL	Tool Command Language
TO&E	Table of Organization and Equipment
UAV	Unmanned Aerial Vehicle
UGV	Unmanned Ground Vehicle
UML	Unified Modeling Language
UV	Unmanned Vehicle
WME	Working Memory Element
XML	Extensible Markup Language