2004 Command and Control Research and Technology Symposium
The Power of Information Age Concepts and Technologies

Proposed Topic Area: C2 Human Factors Engineering

An Intelligent Interface-Agent Framework for Supervisory Command and Control

Dr. Scott D. Wood, swood@soartech.com
Jack Zaientz, jzaientz@soartech.com
Jonathon Beard, beard@soartech.com
Dr. Richard Frederiksen, rdf@soartech.com
Sean Lisse, lisse@soartech.com
Jacob Crossman, jcrossman@soartech.com

Soar Technology, Inc.
3600 Green Court Suite 600
Ann Arbor, MI 48105
734-327-8000

Dr. Marcus Huber, marcush@marcush.net
Intelligent Reasoning Systems
4976 Lassen Dr.
Oceanside, CA 92056
760-806-1497

POC: Dr. Scott D. Wood, swood@soartech.com
Soar Technology, Inc.
3600 Green Court Suite 600
Ann Arbor, MI 48105
Voice: 734-327-8000
Fax: 734-913-8537

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **2004**	2. REPORT TYPE	3. DATES COVERED **00-00-2004 to 00-00-2004**

4. TITLE AND SUBTITLE **An Intelligent Interface-Agent Framework for Supervisory Command and Control**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Soar Technology Inc,3600 Green Court Suite 600,Ann Arbor,MI,48105**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

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

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**		**25**	

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

An Intelligent Interface-Agent Framework for Supervisory Command and Control

ABSTRACT

The Army's vision of the future for armored and mechanized military structure includes the use of mixed teams of human and robotic forces on a dynamic and rapidly changing battlefield. Successful implementation of this vision will require autonomous and semi-autonomous robotic forces and a command and control infrastructure that will allow human, robotic, and mixed teams to be controlled quickly and easily. For maximum effectiveness this infrastructure should allow human commanders to control the robot teams in a similar manner to how they command human teams, that is, in the language of the military, not the language of robotic control theory. Furthermore, the human interface for robotic command and control must simplify warfighter tasks and automate processes such that cognitive workload is reduced, situation awareness is enhanced, and situational control is preserved. In this paper we present initial results from ongoing efforts in developing an intelligent user interface for controlling mixed elements of manned and robotic forces. We have developed a C3 framework of cooperative interface agents that reflect roles found in military command staffs to create a virtual staff for the commander of robotic forces by embedding these military functions within the C3 interface.

An Intelligent Interface-Agent Framework for Supervisory Command and Control

Scott D. Wood
Jack Zaientz
Jonathon Beard
Richard Frederiksen
Sean Lisse
Jacob Crossman
Soar Technology, Inc.
3600 Green Court Suite 600
Ann Arbor, MI 48105
734-327-8000
swood@soartech.com, jzaientz@soartech.com, beard@soartech.com, rdf@soartech.com,
lisse@soartech.com, jcrossman@soartech.com

Marcus Huber
Intelligent Reasoning Systems
4976 Lassen Dr.
Oceanside, CA 92056
760-806-1497
marcush@marcush.net

Abstract

The Army's vision of the future for armored and mechanized military structure includes the use of mixed teams of human and robotic forces on a dynamic and rapidly changing battlefield. Successful implementation of this vision will require autonomous and semi-autonomous robotic forces and a command and control infrastructure that will allow human, robotic, and mixed teams to be controlled quickly and easily. For maximum effectiveness this infrastructure should allow human commanders to control the robot teams in a similar manner to how they command human teams, that is, in the language of the military, not the language of robotic control theory. Furthermore, the human interface for robotic command and control must simplify warfighter tasks and automate processes such that cognitive workload is reduced, situation awareness is enhanced, and situational control is preserved. In this paper we present initial results from ongoing efforts in developing an intelligent user interface for controlling mixed elements of manned and robotic forces. We have developed a command, control, and communications (C3) framework of cooperative interface agents that reflect roles found in military command staffs to create a virtual staff for the commander of robotic forces by embedding these military functions within the C3 interface.

1. 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 information for steel, 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. These 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 forces and sensors. Addressing these needs in a way that improves performance, rather than adding to warfighter workload, requires the development of significantly smarter control and information systems that can accept delegated tasks, monitor significant events, and process information in a way that can speed the transformation of data into understanding. While there are many possible approaches to developing smarter systems, our focus is on the creation of intelligent human-system interfaces that can function as a layer on top of existing and future warfighter battle command and information systems to simplify and augment warfighter interaction.

There are many challenges in creating such systems, including, understanding operational needs and specific human limitations for which intelligent interfaces can help, conducting the basic research and developing the technological infrastructure necessary to prototype intelligent warfighter interfaces, and integrating intelligent 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, our approach has been to explore a very narrow vertical slice through each rather than exhaustively exploring each level prior to addressing the next. In this way, we hope to demonstrate the feasibility of the concept, and the viability of intelligent warfighter interfaces more generally, to build the foundation for a more comprehensive effort. An additional benefit is that by demonstrating how intelligent warfighter interfaces can be applied in practice, it will enable others to envision new applications in ways not currently imagined.

1.1 Project Overview

We describe this research in the context of two complementary projects: 1) Cooperative Interface Agents for Networked Command, Control and Communications (CIANC3), and 2) Battlespace Information and Notification through Adaptive Heuristics (BINAH). Together these projects are addressing the general challenge of developing intelligent user interfaces for military applications and represent two logical sides of such human-system interaction: assisted manipulation of the warfighter's environment (CIANC3) and adaptive delivery and display of information (BINAH). The common idea behind both projects is to create technologies that can reason over multiple knowledge sources such as world events, doctrine, and warfighter task requirements to simplify complex problems, perform routine and delegated tasks, and present information necessary for key decisions at the right time and in the right form.

The focus of the CIANC^3 project has been to identify human-system interaction issues, design potential solutions and create intelligent agent software that support the commander's tasks and mitigates human performance limitations in the context of robotic command and control. The Army's vision for Future Combat Systems includes the use of mixed teams of human and robotic forces on a dynamic and rapidly changing battlefield. Implementing this vision will require a shift from manual, human control of weapons systems to semi- and fully autonomous control over mixed systems of humans and non-human entities. It will also entail an overall force reduction that will require multiple entities to be controlled by individual team leaders and multiple teams to be lead by higher-echelon commanders. To accomplish this, systems will have to be designed to require less human interaction and greater robotic autonomy. Successful implementation of this shift will require autonomous and semi-autonomous robotic forces and a command and control infrastructure that will allow human, robotic, and mixed teams to be controlled quickly and easily. One key to this will be the degree to which teams and individual robots are autonomous. A second is whether the commander's human-machine interface is designed such that the commander is not overloaded with constant system interaction to allow him or her to focus on the mission. We have 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 (simulated in either the JSAF or OTB environments), we have been 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 is to develop software techniques and technologies that will allow human commanders to control the robot teams in a similar manner to how they command human teams, that is, in the language of the military, not the language of robotic control theory.

The focus of the BINAH project has been to improve warfighter capacity to assimilate information in dynamic, high-tempo environments by customizing information delivery based on the context of the warfighter's cognitive readiness, current task, and required decisions. This research is being conducted in the context of Time Critical Targeting, focusing on ground track detection and identification and is applicable to a much wider range of information operations and other knowledge-intensive tasks. For example, the Network-Centric Warfare concept being implemented as part of Department of Defense's Transformation Initiative calls for access to an increasing volume of data and information. Conjoint to the goal of more information is that decisions and operational control will be pushed further down the chain of command, allowing for self-synchronization of battle operations (Power to the Edge). To be effective, such capabilities and responsibilities must be implemented in a way that simplifies and informs the decision-making process without overloading the warfighter with too much information. Here also, we are implementing a suite of intelligent agents and other technologies to fuse heterogeneous data types and to reason about data using temporal, spatial, and causal relationships.

1.2 Warfighter Need for Intelligent Interfaces

In observations of warfighter interaction and other research using prototype battle command systems (e.g. Lickteig, Sanders, Lussier, and Sauer, 2003), we have identified several key areas where some form of intelligent automation might be useful. Broadly these can be divided into two categories: manipulating the environment (command and control) and understanding the environment (situational understanding). 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. Performing this transformation from intent to action can be very direct with experienced warfighters, but automating that transformation 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 replanning (or plan adjustment) is near-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 our efforts to address the warfighter need to manipulate the environment.

Our second focus area, understanding the environment, presents additional challenges to current and future warfighters. Understanding the battle situation, that is, the available information, what that information means to the current situation, and how warfighter actions will affect the course of the mission, are of course key to making informed and effective adjustments to any plan, or to inform the decision-making process of others. By understanding the situation, warfighters can better prioritize decisions, which types of new information are needed for those decisions and in which forms that information will make the most sense. Our experience with intelligence analysts suggests that a large portion of analysis time is spent simply manipulating the data into a form where it can be used. As technology enables more information to flow directly to decision-makers and doctrine is pushing decision-making to lower levels of command, the time and knowledge necessary to perform such manipulations will not be available when they are most needed, that is, during combat. Providing an overwhelming amount of data and information without the ability to automatically transform it will likely result in information overload and other forms of human error (e.g. see Reason, 1990) that can dramatically reduce warfighter performance. This overall task of understanding the situation can be divided into several components, knowing what data to gather, where to find such data, how to transform that data into a form of information that can be useful, knowing how heterogeneous data types can be combined and fused into useful information, and which specific pieces of information are most important. In a static environment where the tasks are constant and the information types well-defined, this could be a fairly straightforward challenge – in a dynamic environment such as battle command where missions, tasks, priorities, and the enemy are constantly changing, analyzing and understanding the situation is difficult and time-consuming. As with transforming commands however, human analysts are often able to adapt very quickly to such dynamic situations if given sufficient time and assistance. Developing the technology and science base to automate information delivery in contextually relevant ways is the primary operational focus of our efforts to help warfighters more easily understand the battle situation and improve decision-making.

In current operational environments, human experts are used to solve the challenges described, by bringing to bear years of experience and knowledge of their tasks. 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 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. Our approach is to encode expert knowledge into forms that can be combined dynamically in the form of intelligent user interfaces and applied in a wide variety of circumstances and purposes. Key to developing such intelligent solutions, systems that augment rather than hinder human performance, is developing a deep understanding of how humans interact with automated and intelligent systems.

1.3 Human-Machine Interaction and Intelligent User Interfaces

The overall goal of the human-machine interface design for this project is to maximize human performance by creating a system that allows users to perform military tasks without focusing on the specific system being used – that is, to allow them to focus on the military objectives rather than on the technological means for accomplishing those objectives. This requires a system that is highly usable: efficient to use, easy to learn, easy to remember, error-tolerant, and subjectively pleasing[1] (Brinck, Gergle, and 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 in ways similar to that of the physical world (e.g. Shneiderman, 1997). 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 to improve efficiency by automating mundane and time-consuming tasks. Previous efforts at automating system tasks have achieved mixed results often because supervisory control issues (Leveson, 1995; Sheridan, 2000) were not adequately addressed. Effectively automating system functions requires achieving a delicate balance between reducing tedious tasks and overall operator workload, and maintaining adequate human control (both real and perceived) and vigilance. For example, users will 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 they need 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 will include the right mix of human and automated control (Wood, 1999; Wood and 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 apply artificial intelligence techniques to every 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), would either engage in a dialog where the system asked a series of questions, or would prepare the set of available data such 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 and Masarie, 1990). Traditional expert systems suffered from three key flaws. First their

---

[1] The degree to which an interface is subjectively pleasing and the criteria by which this quality is measured by users are highly dependent on the domain. In computer gaming, information overload and challenging interaction often adds to a game’s appeal – characteristics that are, of course, not desirable for safety- and mission-critical systems. For complex systems, user affect can probably be more accurately characterized as the degree to which a system fits the user’s mental or physical work style and how it improves the user’s ability to effectively perform his or her fundamental (non-computer) goals.

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, and 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 palpable 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, and 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 helping to overcome data overload and manage complexity. Examples include aerospace fault management systems (Malin et al., 1991) and next-generation medical reference systems (Miller et al., 1986).
>
> Intelligent Interfaces as Elements of Cooperative Systems – Cooperative system elements includes 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 to aid visualization, often from different sources and represented 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 cognitive tools assist with understanding the model, cooperative elements assist with controlling the system and manipulating the model, and representational aids assist with viewing and perceiving relevant aspects of 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, 1998). One way of implementing intelligent user interfaces is through intelligent human-interface agents.

1.4 Why Agents

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

1.4.1 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 C^3 structure will need to adapt to changes in doctrine over time and by service and operation. As requirements change, agents encapsulating the new rules can be introduced into the system without impacting on other aspects of the system.

1.4.2 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 our current research utilizes 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.

1.4.3 Communication oriented design

It is also 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.

1.4.4 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.

1.5 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.

1.6 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, 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 UV operator and the possibility of combat losses, the static and dynamic organization and role determination (Corkill, 1983; So, 1994; So, 1997) will be important issues to address.

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 level (FIPA, 2000; Labrou, 1996; Cohen, 1990; 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 regular, standard patterns and this holds true for intelligent agents as well. These interaction patterns simplify agent reasoning by constraining agent behavior and facilitate creation of expectations and standard behavior models of other agents. Capturing these patterns, commonly called conversation policies or interaction protocols (Bradshaw, 1997; FIPA, 2000; Kumar 2001; Labrou, 1997), is required in any complex multi-agent environment and needs to reflect, for example, any authority relationships that exist between agents (Jones, 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, 1998), whether tasks are distributed or can be handled by a single agent (Gasser, 1990), the level of robustness and fault tolerance required in the domain (Kumar, 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, 1995; Rosenschein, 1985).

2. Agents for Command and Control

To fulfill these requirements we have created a framework of cooperative interface agents based on the roles found in current command staffs. Command staffs commonly provide five basic functions to commanders in support of reconnaissance, security, offensive, and defensive operations (e.g. FM 6-0, 2003):

- 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.

In $CIANC^3$ these functions are divided between three classes of agents: tasking, monitoring, and coordinating. This division aligns the agents with the three C^3 concepts of command, control, and communication respectively. Notionally, interface agents would 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 will remain. In addition, it is assumed that interface agents will have access to, and be integrated tightly 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 remain. These agent classes are discussed below with examples of how they might be used.

2.1 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 should engage the commander in dialog to clarify intentions or will present several options. After customizing the resulting plan as necessary, the commander can then issue the order. The tasking agent will 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 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 should be visible to the commander so that they can be inspected, monitored, critiqued, and modified. This mix of interface agent assistance and direct manipulation is essential to achieving the right mix of automated and manual control. Examples of other roles tasking agents might play 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.

2.2 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, IFF and communications security are all time-consuming but necessary issues with which coordinating agents will be able to assist.

For example, 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 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 example where coordination is critical is rapidly responding to fast-moving or stealthy targets. Coordinating air defenses and sensor systems faster than humanly possible is often necessary for effectively countering such attacks. In such situations, the coordination agent might work directly with monitoring and tasking agents to rapidly eliminate the threat. Other roles that might be played 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., tele-operation) in critical situations.

2.3 Monitoring Agent

Monitoring agents are responsible for assisting the commander in maintaining 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 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 such as 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, especially when the commander is engaged in critical tasks, to minimize distractions. For example, if the commander is busy responding to an ambush with 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 current commander tasks. For instance, message traffic and information flow may increase dramatically during a firefight. Where loss of life or equipment is imminent, relevant information that might prevent or mitigate the situation could be made 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 COP (common operational picture) 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.

2.4 Specialist Agents

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

2.4.1 Network Effects Agents

Network Effects agents are responsible for responding 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; ontological queries to build an inference-based understanding of platforms, weapon systems, and targets; determination of the feasibility of employing particular platforms and weapon systems against particular targets; employment of requested effects against requested targets; and requests for maneuver of particular platforms and weapon systems into configurations more suitable to employment of effects. By abstracting effects requests away from specifically identified platforms and weapon systems, the formation of ad hoc teams on demand reduces both kill-chain latency and commander workload overhead. These features mean that Network Effects agents can significantly contribute to assisting the increase of operational tempo for battlefield commanders.

2.4.2 Network Sense Agents

Analogously to Network Effects agents, Network Sense agents are responsible for responding 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; ontological queries to build an inference-based understanding of platforms, sensor systems, areas of interest, and targets; determination of the feasibility of employing particular platforms and sensor systems against particular targets; employment of sensors to obtain requested information; and requests for maneuver of particular platforms and sensor systems into configurations more suitable to gathering of sensor information. By abstracting sensor requests away from specifically identified platforms and sensor systems, the formation of ad hoc teams on demand reduces kill-chain and battle damage assessment latency, and commander workload overhead. These features mean that Network Sense agents can significantly contribute to assisting the increase of operational tempo for battlefield commanders.

2.4.3 Network Maneuver Agents

The purpose of the Network Maneuver agent is, upon request, to direct particular platforms to engage in particular maneuvers on the basis of platform capability descriptions. For example, this enables a Network Maneuver agent to respond to a request that a platform with an anti-tank capability and I/R sensing capability maneuver to a particular location (perhaps in response to a platform maneuver request generated by a Network Effects or Network Sense agent). Checking available resources to see which platforms might have these capabilities, when making platform selection decisions the Network 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 best match platform for a particular request. Again, these features mean that Network Maneuver agents can significantly contribute to assisting the increase of operational tempo for battlefield commanders.

2.4.4 *Information Fusion Agents*

Information Fusion agents are responsible for monitoring data streams and performing value-added data transformations. Depending on data stream, and the degree it has already been processed, these transformations can include entity identification, situation assessment, and threat assessment. While monolithic and centralized data fusion systems will continue to play an important role, they lack the ability to focus fusion efforts, and bring appropriate knowledge to bear, on local problems based on individual warfighter current context, behavior, workload and training. In addition to directly supporting the warfighter, Information Fusion agents can act as a key system service providers, managing and responding to information requests from other system components. Information Fusion agents will help provide decision quality information to the warfighter, including correlating track data reports from heterogeneous sensor types, identifying tracks that do not conform to expected behavior, provide estimate threat levels, prioritize tracks of interest and identify areas where incoming data reports are not meeting ISR requirements or where sensor capabilities have changed.

2.4.5 *Display Control Agents*

Display Control agents are responsible for monitoring the warfighters display environment, COP, and current cognitive readiness and for configuring the displays to maximize the warfighter's ability to absorb relevant information. The warfighter's ability to absorb and make decisions based on information displays can vary greatly depending on their physical and mental condition, training, workload, and operational environment. While in many circumstances, information can and should be tailored to these conditions, standard information displays have limited capability to react to these changes and little ability to change display characteristics in response to them. Display Control agents will help maximize the warfighters awareness and decision making ability by switching key information between visual to audio displays, highlighting priority or recently changed information, and highlight relationships between data.

2.4.6 *Data Channel (Stringer) Agents*

Stringer agents assist the warfighter in maintaining a managed connection to heterogeneous sensors and data sources. Each Stringer agent is designed to understand the details, protocols and data formats of a particular data feed. Stringer agents serve two roles, first to translate tasking agent tasks to specific resource instructions, monitoring communication across their data feeds and filtering, translating, and preparing the raw data for use. Secondly, the Stringer agent acts as a transducer, translating raw communication into a form usable by the agent system. In a Time Critical Targeting system, for example, Stringer agents could be created to deal with a variety of targeting reports from satellites, ground and air radar and field reconnaissance as well as provide linkages to the COP and other C2 systems.

3. Implementation Environment

The current CIANC^3 system integrates Soar-based interface agents into a combined simulation and operational environment for robotic control. The agents communicate using a FIPA compatible ACL and a user interface to the agents was created using TCL and Java.

3.1 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, RWA, related controllers and more recently to model ground forces (Taylor, et al., 2001). For example, we have created 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.

3.2 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 will enable robotic forces to perform these types of communication, but the most applicable is that based on Joint Intention (JI) theory (Cohen and Levesque, 1990; Huber et al 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 will enable 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.

3.3 Protégé/Ontology Approach and DAML2Soar

We are currently focusing ontology representation solutions on complete ontology representations in agent memory (CM). Most existing ontologies remain modestly sized and representing these ontologies directly in memory does not adversely impact performance in Soar. This solution also allows us to explore incremental transition of the ontology to long-term memory via Soar's native learning mechanism. We have 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. 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. Thus, 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 this approach and to explore alternatives to it.

4. Results

Our current effort has focused on developing the fundamental architecture to demonstrate viability of an agent-based approach to supervisory command and control and to facilitate continuing research. To do this we have developed a very narrow set of functionality for a limited operational scenario. As planned, this approach has resulted in a modest demonstration of new capabilities, yet has made apparent many of the inherent challenges in implementing network-centric solutions (independent of whether the approach is agent-based). The CIANC3 project was conducted in the context of the Army's Future Combat Systems (FCS) program, tasked to explore agent-based technologies for systems that do not yet exist and for doctrine that has not been fully developed.

Our working scenario was based on the FCS Unit of Action Baku vignette and the demonstration prototype was designed to provide entity level control and coordination based on commander Operational Orders (OpOrders). In our demonstration, the system reasons over simulated entity capabilities and disposition, rules of engagement, the current operating scenario, and commander's intent – to task and coordinate networked sensor, maneuver, and effects in real time. Figure 1 shows the layout of the working scenario.

Figure 1. In the demonstration scenario a CIANC^3 controlled FCS company in the upper left of the picture assaults a red force compound in the lower right of the picture. The scenario is currently implemented using the JSAF simulation environment.

In this scenario an 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.

In executing this scenario, our 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. Agent infrastructure capabilities include

1) Arbitrary sets of simulated Blue Force entities and their capabilities can be registered with, and accessed from, a prototype Directory Service.
2) The Monitoring Agent can request, receive and propagate status messages from all entities registered with the directory services.
3) The Tasking Agent can dynamically assemble Blue Force teams based on the commanders plan requirements and to establish system goals, subgoals, and rules of engagement derived from the commanders plan.
4) The Coordinating Agent can provide detailed instructions to Blue Forces and monitor for task completion or interruption and react to plan interruptions.

The demonstration prototype is still very 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 necessary reflect optimal tactics. Some specific tactical scenario abilities include:

1) The system takes a general request for UAV sensor platform to perform reconnaissance and identify and tasks specific assets.
2) The system can react to loss of a UAV asset, noting the disruption of the plan and assigning a new asset to the task.
3) The system can assign assets to routes and issue fire requests and ROE changes.

Looking at how the system assigns a UAV to a recon point provides a good example of how the agent framework operates. The initial battle plan includes an area or point to be reconnoitered and a general description of sensor type required. The Tasking agent, by querying Directory Services and the Domain Ontology, identifies a specific UAV that is available and has the desired sensing capabilities. Then the Tasking Agent communicates with the Coordinating Agent, informing it of the goal to recon the point with the specific asset. The Coordinating Agent (eventually supported by the Maneuver and Sensor agents) then issues specific movement commands to the UAV. The UAV move to position, reporting 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).

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 maneuver, incorporation of 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.

5. Discussion

As a result of the research described here, we can make some initial claims about novel aspects of our work that can inform future efforts and contribute scientifically, technically, and operationally.

5.1 Design of Multi-Agent Teams for Mission-Critical Applications

Much of our effort to date has gone towards creating the technical infrastructure that will permit more in-depth research into how intelligent warfighter interfaces can best be used. 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, we claim 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. We discuss these claims further in the following sections.

5.1.1 Agent Communication Languages

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, the intended result of the speaker, is differentiated from the perlocutionary force, 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 and goals and intentions are quite natural ascriptions by humans to each other and to complex systems in general. It is this intentional stance (Dennet, 1986) that permits us 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 belief, goals, intentions, etc. The Soar agent architecture naturally supports ACL definitions. 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 and Levesque, 1990b) embodies a persistent commitment to act on a particular goal, which Soar also naturally captures in its operator execution framework.

We use a variant of the Agent Communication Language semantics defined by Cohen and Levesque and extensions (Cohen and Levesque, 1990a; Cohen and Levesque, 1990b; Cohen and Levesque, 1990c; Cohen and Levesque, 1991a; Cohen and Levesque, 1991b; Cohen and Levesque, 1995; Huber et al., 2001; Kumar et al., 2000; Kumar et al, 2002; Smith and Cohen, 1996). The semantics will be extended to included deontic modal operators.

5.1.2 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 $O_x a$ (or (OBLIGATED x a)) says that the agent x (often left unspecified, i.e., Oa or (OBLIGATED a)) is obligated to perform action a and is taken to be a primitive in many formal theories of deontics (e.g. (von Wright, 1951; Horty, 1993; Jones and Sergot, 1996)). However, with respect to this project, we tie this in formally with the "Joint Intention" theory described in the Agent Communication Language (ACL) article associated with this document. By formally conjoining these two semantic theories, we gain the following significant advantages:

- 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, we see that the agents are required to perform an action then (the ramifications of the obligation) to reach mutual belief regarding success or failure.
- A specification of to whom the agent is obligated. While an agent may be thought as becoming obligated to itself at some point in time (a form of intention, perhaps), the interesting aspect for the CIANC[3] project is the obligations incurred between agents. Because of this, we will define OBLIGATED with respect to whom the agent is obligated. I.e., (OBLIGATED x y a) will indicate agent x is obligated to do action a for agent y.
- A unified semantics addressing both a deep and rich intentional utterance semantics with the deontic aspects of obligations and permissions, both of which are 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 been well unified into a single cohesive, semantic framework.

5.1.3 Supporting Work

In support of these claims we have defined a single, coherent set of basic semantic and notational definitions underlying joint intention theory. JI definitions have changed slowly over time in the research, both semantically and notationally as limitations are eliminated or extensions made and this can be confusing when piecing together a set of ACL performatives. We have defined a single, coherent set of performative definitions. Prior research efforts led to narrowly focused redefinitions of performatives in the literature as the basic underlying definitions changed. However, not all performatives previously defined in the literature 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 again, ostensibly to remove limitations, provide extensions, etc. Finally, we have defined a broad, "complete" set of performative definitions. Not all of the performatives that might be considered necessary for fielding a multi-agent system have been previously defined in the literature, notably "utility" performatives, both those implicitly required by joint intention theory and those not so required but found to be useful when fielding systems based on ACLs with other semantics.

5.2 Knowledge Representation and use in Multi-Agent Systems

A stated goal of the US armed forces is to greatly increase its warfighting effectiveness through the use of unmanned and computer augmented systems such as unmanned vehicles, intelligent interfaces, and command and control assistants. It is well understood that significant increases 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. We claim that any highly capable system for assisting warfighters in battle command functions will need to solve this challenge in a general way. Furthermore, once this challenge is solved for one function (e.g.

operations), the resulting knowledge can readily be applied to any number of related functions (such as training, planning, analysis, etc.).

An implicit 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 provide a means for adjusting their behavior in a way that is easy to understand and simple enough to do in a short timeframe.

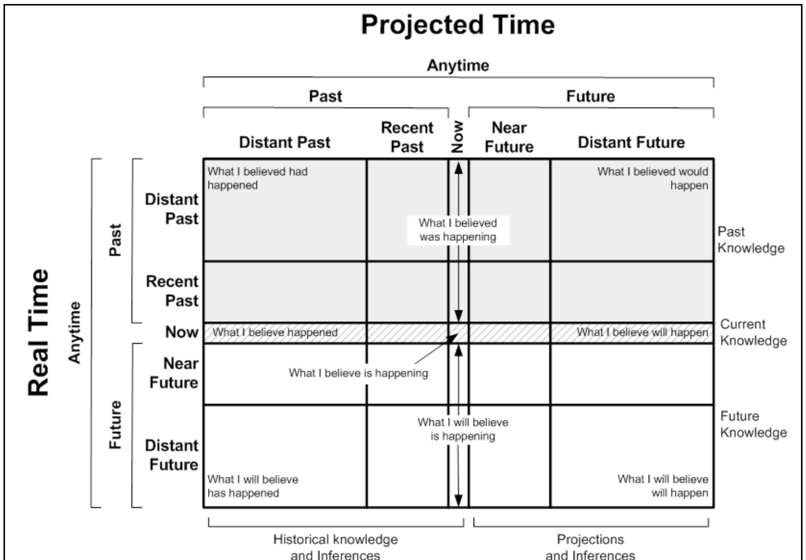

Figure 2. A graphical representation of 2-dimensonal time.

It has been shown (Trafton, et al, 2003) that humans are effective at thinking about complex problems qualitatively. To simplify the necessary reasoning and facilitate understanding, we structure our formalism around qualitative terms and reasoning, borrowing from temporal database theory (Snodgrass and Ahn, 1983) and qualitative process theory (Forbus, 1984). The first part of this formalism is a qualitative representation of time, space, and uncertainty. For example, Figure 2 presents our two-dimensional representation of time that allows a broad class of questions to be answered about events occurring in time, including when the agent thought about the events.

The exact number of qualitative bins may vary as required. Furthermore, the exact boundaries of these intervals can be modified to reflect the temporal scope of the inferences and the task to be accomplished. Using the above time representation, we can reason about what is happening now, what happened in the past, what will happen in the future, and what was previously thought about.

The second part of this formalism is a pseudo-English language grammar for encoding inferences. The language is grounded in the entities the system knows about as well as the qualitative representation of space, time, and uncertainty within which the information is known and thought about. An example heuristic might be in a situation where the system is monitoring intelligence traffic about new contacts in the area of operations:

> Heuristic: If now or in the past I did not believe I knew about this potential target and this potential target is not tagged new, then this potential target is tagged new now.

The inference system can also be used as a reasoning engine behind an intelligent user interface. For example, one job of the system would be to monitor the user's task and workload to be able to manage the information presented to the user. An example heuristic along these lines might be the following:

Heuristic: If now I believe user workload is high, and there is a contact C1 with priority low, then reduce saliency of C1.

Our approach is an important step toward autonomous decision making over time in a way that is both sufficient for most reasoning tasks and yet understandable to non-programmers. The formalism we have developed can benefit many systems by providing a foundation on which to develop intelligent user interfaces, autonomous unmanned vehicles, robotics command and control, intelligent data aggregation and filtering, and intelligent digital assistants. Furthermore, by being understandable to non-programmers, our formalism provides an excellent medium for encoding user knowledge in a form that can be both inspected and executed.

5.3 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."

To meet this demand requires systems that, 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, et al. 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, we believe 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 our approach have been to work towards increasing the warfighter's span of control for human-robot interaction and improving workload management. Current state-of-the-art has multiple personnel controlling a single unmanned platform. Our approach centers on enabling a 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 requires 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 C4ISR roles, responsibilities, and capabilities is central to this approach, and leverages the knowledge-rich character of the agents. To do this we rely on the agents' ability to access and reason over the knowledge sources listed above as well as others. This provides numerous benefits including,

- agent behavior that is more comprehensible and explainable to potential field users than 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, we are 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 of knowledge-intensive agents can assist in information processing and robotic platform control for both commanding officers and robotic controller NCOs by referencing shared knowledge repositories and providing 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 simplify command and control system development, enabling knowledge required in multiple sub-systems to be encapsulated and shared, and allow common agent capabilities to be used in multiple contexts.

6. Conclusions

This paper describes ongoing efforts to develop agent-based intelligent user interfaces for battle command and intelligence analysis. 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. Our techniques for defining and using 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 we've developed will be necessary for any robust multi-agent system.

Acknowledgements
The work described here was conducted under a Department of the Army Small Business Innovation Research Program 2000.2 contract for topic A02-024 (contract #DASW01-03-c-0019) and an Office of Secretary of Defense Small Business Innovation Research Program contract administered by Air Force Research Laboratories – Information Directorate (contract # F30602-03-C-0022. We gratefully acknowledge the sponsorship of this research by the United States Army Research Institute and the Air Force Research Laboratory, and thank ARI COTR Carl Lickteig and AFRL COTR Robert Hawkins for their guidance and support.

7. References

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

Brinck, T., Gergle, D., and Wood, S.D., 2001. Usability for the Web: Putting Usability First. Morgan Kaufmann: San Francisco.

Cohen, P. R. and Levesque, H. J., 1990a, "Rational Interaction as the Basis for Communication". In P. R. Cohen, J. Morgan, and M. E. Pollack, Eds. *Intentions in Communication*, System Development Foundation Benchmark Series, MIT Press, Cambridge, MA, 221-256, 1990.

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

Cohen, P. R., and Levesque, H. J., 1990c, “Performatives in a Rationally Based Speech Act Theory”, In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, Pittsburgh, Pennsylvania, 1990.

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

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

Cohen, Philip R. and Hector J. Levesque, "Communicative Actions for Artificial Agents", in *Proceedings of the First International Conference on Multi-Agent Systems*, San Francisco, CA, 65-72, 1995.

Daniel C. Dennett, *The Intentional Stance*, MIT Press, 1987.

FM 6-0. (2003). Mission Command: Command and Control of Army Forces. August, 2003.

Horvitz, B. *Principles of Mixed-Initiative User Interfaces.* In Proceedings of CHI'99, ACM SIGCHI Conference on Human Factors in Computing Systems, May 1999.

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

Huber, Marcus, and Kumar, Sanjeev, and Cohen Philip R., and McGee, David R., “A Formal Semantics for Proxy Communicative Acts”, In *Proceedings of Agent Theories, Architectures, and Languages (ATAL) 2001*, Seattle, WA, August, 2001.

Huhns, M. N., & Singh, M. P. (1997). Agents and Multiagent Systems: Themes, Approaches, and Challenges, *Readings in Agents* (pp. 1-23). San Francisco, CA: Morgan Kaufmann Publishers, Inc.

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

A.J.I. Jones and M.J. Sergot. “A Formal Characterisation of Institutionalised Power”. *Logic Journal of the IGPL*. 4(3):429-445, June, 1996.

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

Kumar, Sanjeev, and Huber, Marcus, and Cohen Philip R., “Representing and Executing Protocols as Joint Actions”, In *Proceedings of the First International Conference on Autonomous Agent and MultiAgent Systems (AAMAS)*, pgs 543-550, Bologna, Italy, June, 2002.

Kumar, Sanjeev and Huber, Marcus, and Cohen Philip R., and McGee, David R., “Toward a Formalism for Conversational Protocols Using Joint Intention Theory”, *Journal of Computational Intelligence (Special Issue on Agent Communication Language)*, Brahim Chaib-draa and Frank Dignum (Guest Editors), Vol. 18, No. 2, pages 174-228, 2002.

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.

Lieberman, H. (1997). Autonomous Interface Agents. In *Proceedings of the ACM Conference on Computers and Human Interaction [CHI-97]*, Atlanta, March 1997: ACM Press.

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

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.

Sheridan, T. B. (2000). Function allocation: Algorithm, alchemy, or apostasy? *International Journal of Human-Computer Studies, 52*, 203-216.

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

Taylor, G., Koss, F., Nielsen, P. (2001). Special operation forces IFORs. Proceedings of the Tenth Conference on Computer Generated Forces. May, 2001.

U.S. Department of the Army (1991). *Cavalry Operations* (FM 17-95). Washington DC.

G.H. von Wright, "Deontic Logic", Mind. 60:1-15, 1951.

Wray, R., Lisse, S., Beard, J. (2004) *Investigating Ontology Infrastructures for Execution-oriented Autonomous Agents*. Paper presented at the AAAI Spring Symposium 2004.