Using Intelligent Agents to Control Teams of Robotic or Simulated Entities

Scott D. Wood, Jack D. Zaientz
Soar Technology
Ann Arbor, MI
swood@soartech.com, jzaientz@soartech.com

ABSTRACT

DARPA'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 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. This infrastructure will need to 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 the controller's tasks and automate processes in such a way that the cognitive workload is reduced, situation awareness is enhanced, and situational control is preserved.

In this paper we present a theoretic approach to creating such a system. The approach is centered on recasting the robotic control problem as a supervisory control problem, focusing on the human commander as supervisor of an intelligent command and control (C2) system. To explore and evaluate this approach our team is building a C2 framework of cooperative interface agents that reflect roles found in current command staffs. The intention is to create a virtual command staff for the robotic controller by embedding these military functions within the system itself. Although this work is ongoing, the results to date suggest that the approach we have taken will significantly reduce cognitive and human performance workload, thus improving overall usability and reducing training requirements. In addition to robotic control, this work opens up a new approach to controlling simulated entities during experimentation, training and mission rehearsal.

ABOUT THE AUTHORS

Scott D. Wood, Ph.D., is a Senior Scientist with Soar Technology since February 2001. Dr. Wood has over ten years of research and industry experience in the areas of software development, e-business consulting, cognitive modeling, and human-computer interaction. His doctoral research included extending GOMS (Goals, Operators, Methods, Selection Rules) modeling to allow for human error, developing techniques for predicting where human errors would occur in an interface, and testing those techniques by applying them to web applications. Dr. Wood also has extensive experience developing human-performance models using the EPIC cognitive architecture, optimizing workflows and interface usability through task analysis, and in designing web solutions for e-business applications. In addition, he spent four years in the U.S. Army attached to 5th Special Forces Group and other special operations units. He earned a B.S. in Computer Science (1990) from Tulane University, and M.S. (1994) and Ph.D. (2000) degrees in Computer Science and Engineering from the University of Michigan, Ann Arbor.

Jack D. Zaientz, is a Research Scientist with Soar Technology since August 2001. Mr. Zaientz has over eight years of experience in Human-Computer Interaction, analyzing, designing and implementing applications and application interfaces for a wide range of industries including financial, legal, insurance, commercial, scientific, military and intelligence service. His current research and design work focuses on Human-Computer Interaction issues involving interface design, information visualization and agent systems. He has a background in cognitive psychology and cognitive modeling and, prior to coming to Soar Technology, developed cognitive modeling tools for the Cognition Lab at NASA's Ames Research Center. He earned a Masters of Human Computer Interaction from Carnegie Mellon University's Human Computer Interaction Institute (2001).

Using Intelligent Agents to Control Teams of Robotic or Simulated Entities

Scott D. Wood, Jack D. Zaientz
Soar Technology
Ann Arbor, MI
swood@soartech.com, jzaientz@soartech.com

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, 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 for battle command and simulation control 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.

In addition, JV2020 calls for these capabilities to be developed within a process of continuous of experimentation and simulation to provide feedback into the design of these capabilities and to allow the Services and combat commands to use these technologies to explore changes in doctrine, organization, materiel, and training. This process of continuous learning will help the US military choose innovations that take greatest advantage of new ideas and technologies to field improved capabilities for the joint force. This process, though, will require similar improvements in capabilities in order to cope with complexity of controlling the future scenarios envisioned by JV2020. Simulation designers, controllers and evaluators will the capabilities of a battlefield commander to provide scenarios with battlefield validity and to provide positive warfighter training.

Project Overview

We describe this research in the context of the CIANC^3 (Cooperative Interface Agents for Networked Command, Control and Communications) project. 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 autonomo us. 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.

Warfighters Need for Intelligent Interfaces

In observations of warfighter interaction and other research using prototype battle command systems, we have identified several key areas where some form of intelligent automation might be useful. Broadly these can be divided into 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 subordinates, often only intent is conveyed (or even necessary) from which the recipient can add available context to turn that intent into 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, only compounds this 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. Transforming commands and coordinating multiple unmanned elements is the primary operational focus of our efforts to address the warfighter 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 of their tasks. Developing automated solutions that approach or exceed human capabilities 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 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. Understanding how humans interact with automated and intelligent systems, however, is key to developing intelligent solutions that augment rather than hinder human performance.

Human-Machine Interaction and Intelligent Us er 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 them 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. One technique for maximizing usability is to automate mundane and time-consuming tasks with software. 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 a delicate balance of 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. 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, 1994), 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 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 software interface agents.

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.

Encapsulation of knowledge - Localizing doctrinal knowledge in specialized agents provides a natural mechanism for matching interface-processing rules with military doctrine. CIANC3 agents will be part of the DOD's C3 structure, and as such 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.

Encapsulation of Processing - Localizing task execution in specialized agents also provides a natural mechanism for encapsulating processing. As the duties of the individual CIANC3 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 quite reasonable to assume that future CIANC3 agents may require the addition of dedicated planners, case-based reasoning systems, or other AI technology.

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. Since the purpose of the CIANC3 system is assisting the commander in the command, control and communication of external robotic entities; a rich capacity for communication is a basic requirement. Establishing this capacity as a fundamental characteristic of the architecture allows the seamless introduction of new processing units at any time or at any location in the CIANC3 architecture.

Reconfigurable design - It should also be assumed that the target agent organization described here would 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 the 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, and 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.

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. FM3-1, 2003): 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 CIANC3 these functions are divided between three classes of agents: tasking, monitoring, and coordinating. This division aligns the agents with the three C3 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.

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.

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

Examples of 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 and tacit 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.

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. 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, representing 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, we have developed an initial set of specialist agent types that are instantiated and applied for specific tasks.

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.

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.

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.

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.

IMPLEMENTATION ENVIRONMENT

The current CIANC3 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.

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. In all cases, human operators can also provide command and control to Soar agents.

Agent Communications Infrastructure

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.

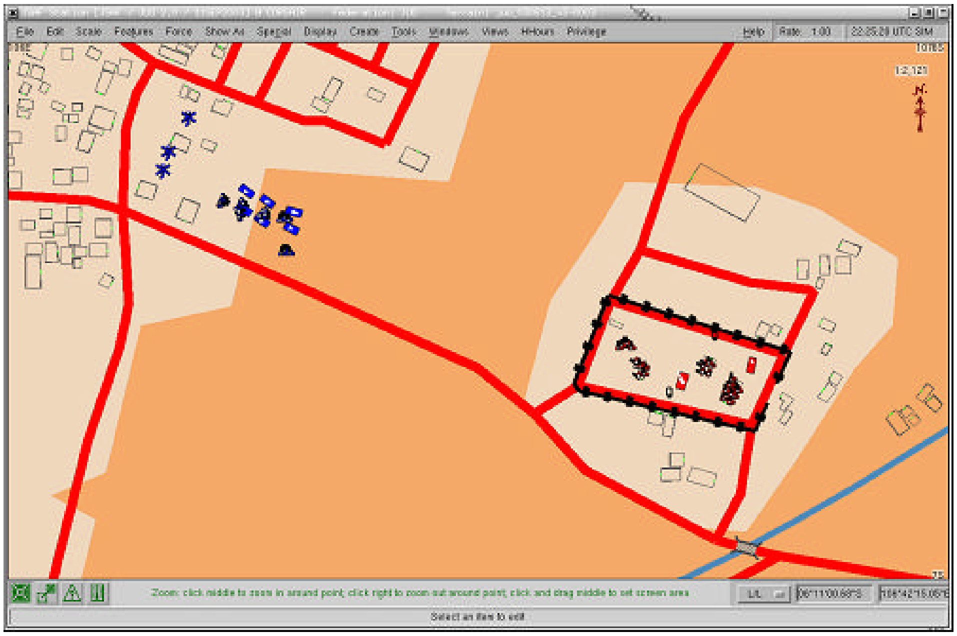
Simulation Testbed

Developing experimental 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. Currently the CIANC3 system has been successfully integrated into both the Joint SAF (JSAF), OneSAF Testbed (OTB and OTB^2) simulation systems. Integrating with these simulation systems has provided CIANC^3 with simulated entities to control and receive status from as well as a tool for creating opposing force behaviors.

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 (below) shows the layout.



Zoom: click middle to zoom in around point; click right to zoom out around point; click and drag middle to set screen area

Select an item to edit

Figure 1. Demonstration scenario in JSAF simulator

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.

DISCUSSION

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.

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. Not specifically identified by Potok, but demanded by the objectives of Joint Vision 2020 and the nature of the military domain, is the requirement that the agent based system be knowledge-intensive with a high degree of problem solving capabilities.

A primary goal of our approach has been to work towards increasing the warfighter’s span of control and improving workload management. Current state-of-the-art has multiple personnel controlling a single uninhabited platform. Our approach centers on enabling a person to control multiple uninhabited 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, and 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.

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 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 battlefield command and could be served by similar supporting technologies.

The intelligent user interface approach we have 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, it has already begun to show its benefits in the simulation arena, providing the infrastructure for improved control mechanisms including; ad hoc group creation and multi-entity tasking, entity and group level reactive planning and status monitoring. The basic infrastructure we have implemented has been successfully integrated with a number of DoD simulation environments and promises to develop into a powerful tool for simulation control.

CONCLUSION

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). We gratefully acknowledge the sponsorship of this research by the United States Army Research Institute and thank ARI Program Manager Carl Lickteig for his guidance and support.

REFERENCES

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

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

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.

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

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

Maes, P. (1994). Agents that reduce work and information overload. *Communication of the ACM, 37*(7), 30-40.

Malin, J. T., Schreckenghost, D. L., Woods, D. D., Potter, S. S., Johannesen, L., Holloway, M. and Forbus, K. D. (1991). Making Intelligent Systems Team Players; Case Studies and Design issues. *Vol 1: Human Computer Interaction Design; Vol 2: Fault Management System Cases. Houston,* TX, NASA Johnson Space Center.

Maybury, M. T., Wahlster, W. (1998). Intelligent User Interfaces: An Introduction. *Readings in Intelligent User Interfaces.* M. T. Maybury, Wahlster, Wolfgang. San Francisco, Morgan Kaufmann Publishers: 1-13.

Miller, R. A., McNeil, M. A., Challinor, S. M., Masarie, M. D. and Myers, J. D. (1986). "The INTERNIST-1/QUICK MEDICAL REFERENCE Project -- Status Report." *Western Journal of Medicine* 145: 816-822.

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*.

Roth, E. M., J. T. Malin and D. L. Schreckenghost (1997). Paradigms for Intelligent Interface Design. Handbook of Human-Computer Interaction. M. Helander, T. K. Landauer and E. Prabhu. Amsterdam, NL, Elsevier Science B.V.**:** 1177-1201.

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

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

Wood, S. D. (1999). The application of GOMS to error-tolerant design. In *Proceedings of the 17th International System Safety Conference*, Orlando, FL.