Integrating Intelligent Agents into Human Teams

Katia Sycara Carnegie Mellon University

Michael Lewis University of Pittsburgh

1. Introduction

As the role of teams becomes more important in organizations, developing and maintaining high performance teams has been the goal of several researchers (Decker, Sycara, & Williamson 1997; Beyerlein, Johnson, & Beyerlein, 2001, Cannon-Bowers & Salas, 1998; McNeese, Salas & Endsley, 2001; Salas, Bowers, & Edens, 2001) . One major question is how to turn a team of experts into an expert team. Several strategies such as task-related cross training (Salas, Cannon-Bowers, & Johnson, 1997; Cannon-Bowers and Salas, 1998) have emerged. In addition to traditional behaviorally-based methods for facilitating team development, advances in computer science and robotics are now allowing the introduction of artificial intelligence (sic; intelligent agents) into teamwork in a variety of roles and functions. This advent of agent technologies raises two questions: 1) what kinds of assistance can be provided and 2) what kinds of assistance prove beneficial. The reported research attempts to answer these questions through technology demonstrations to document what is possible and behavioral experiments to identify what may be helpful.

The greatest impediment to assisting human users lies in communicating user intent to an agent and making the agent's results intelligible to the human. Today in almost all cases the limiting factor in human-agent interaction has become not computing cycles or connectivity (the machine side) but the user's ability and/or willingness to communicate his desires and sift, organize, and interpret the machine's response to satisfy them (the human side). The characteristics of increased flexibility and autonomy that make agents suitable to plan and execute tasks on behalf of human users also make monitoring and evaluating more difficult for the humans. For example, if you were to task an agent to book an inexpensive flight to Athens with a departure on Tuesday you should not be surprised to get back an itinerary with a 14 hour overnight layover in Memphis another in Warsaw and an arrival on Thursday. By the time you have enumerated your preferences in sufficient detail to have confidence in the agent's booking you might as well have gone online and booked it yourself. Except in cases where an agent's task performance is completely correct and deterministic, uncertainties as to agent progress in performing the task, alerting the user to potential failures, or protecting the user from unauthorized agent actions may need to be addressed for even the simplest interactions.

The degree of difficulty of these challenges varies with an agent's role. There are three possible functions that software agents might have within human teams:

1. Support the individual team members in completion of their own tasks

2. Assume the role of a (more or less) equal team member, by performing the reasoning and tasks of a human teammate

3. Support the team as a whole.

The first function is to support the individual team members in completion of their own tasks. This approach focuses on the specific tasks that an individual must accomplish as part of the team and the subdivision of tasks among the human and the agent (or agents) to ensure high quality performance. In a scheduling task, for example, the agent might propose an initial schedule for the human to examine and suggest reordering based on knowledge about the team's goals not considered in the agent's computations. The agent could then prepare a new schedule incorporating the changes in priorities to be approved by the human and forwarded to the team. In this example the agent and human share task responsibility, with the agent providing algorithmic scheduling capabilities and the human supplying more detailed knowledge of team goals. Aiding individual tasks can improve team performance both by improving the individual's performance, in this case a better schedule, and by freeing the human's cognitive resources for teamwork.

With the second function, serving as a teammate, issues associated with communication and coordination among team members become relevant as well (Cannon-Bowers and Salas 1998, Lee and Moray 1992, Lenox et al. 1997, Tambe and Zhang 1998, Grosz and Kraus 1996). A software agent in this role must not only perform its assigned task but must use communications and modelling to share information, goals, and maintain its intelligibility to other team members. If the scheduling agent were promoted to team member status, for example, it would need a much more sophisticated model of the team's goals and interdependencies among its teammates' tasks in order to make the same adjustments to job priorities. Filling the role of a human teammate is extremely challenging for software agents because they are idiot savants, capable of very complex computations but requiring intensive AI programming to replicate the sorts of commonplace reasoning and ad hoc assistance we would expect of a human in the same role.

The third function is to support the team as a whole by facilitating communication, allocation of tasks, coordination among the human agents, and focus of attention. Issues here deal with how to support interactions among team members using agents (Lenox et al. 1998, Lenox et al. 1999), what kind of software agent architecture and processing allows agents to monitor team activity, access and distribute information and results of their reasoning to human team members that need them. Specifically, the focus is on how software agents could be used to support and promote teamwork along the dimensions identified by Cannon-Bowers and Salas (1998). Surprisingly, the task of supporting teamwork explicitly appears more amenable to agent assistance than that of incorporating teamwork into the performance of individual tasks. As part of the communications infrastructure, a software agent can initiate searches for supporting and related information or facilitate passing information to appropriate teammates without the sophistication of modelling needed to fill a human role.

Tasks confronted by human teams often require individual specializations, separated locations, and differing responsibilities. No single agent could provide all the forms of assistance diverse members of a team might need. A system comprised of multiple, functionally specific agents, however, can configure itself as needed to deliver the right support to the team and its members tailored to the team's particular task and situation. For multiple agents to provide effective flexible assistance, however, requires developing a sophisticated multiagent infrastructure with explicit mechanisms for communication and coordination.

We have developed the RETSINA (Reusable Environment for Task Structured Intelligent Network Agents) re-usable multiagent infrastructure (Sycara et al. 1996, Decker et al. 1997) which provides a domain independent, componentized framework to (a) allow heterogeneous multiple agents to coordinate in a variety of ways, (b) enable a single agent to be part of a multiagent infrastructure, and (c) allow effective human-agent interaction. To this end, RETSINA provides facilities for reuse and combination of different existing low level infrastructure components and also defines and implements a sophisticated individual agent architecture that provides higher level agent services, such as planning and execution of various tasks.

In this chapter we report on our research to address the challenges of integrating agent technology in support of human teams. Our research approach consisted of (a) identification of a validated model of human teamwork in high performance teams to guide identification of situations where agents could provide value-added assistance as well as identification of abilities that an agent should have in order to provide the needed assistance, (b) development and extension of our previous work on the RETSINA infrastructure to demonstrate agent-based team aiding in different complex, time-stressed scenarios, such as joint mission planning and Non Combatant Evacuation and (c) evaluation of agent team aiding in controlled laboratory experiments on simpler but still useful problems, namely a target identification task and a path planning task.

The rest of the chapter is organized as follows: Section 2 presents a brief overview of teamwork models, various teamwork challenges, and various teamwork dimensions that we used to base our identification of agent-based team aiding strategies. Section 3 presents a brief overview of the RETSINA infrastructure and an illustrative demonstration scenario, pointing out the different kinds of assistance the agents can provide. Section 4 presents controlled experiments we conducted using the TANDEM synthetic radar task to investigate the need for agent intelligibility and agent support for teamwork. Section 5 presents experiments using the MokSAF route planning simulation to investigate agent impacts on a deliberative team planning task. Section 6 presents conclusions and future work.

2. Models of Teamwork

Our approach to human teamwork is based on the ATOM model proposed by Smith-Jentsch et al. (1998a) Using a principal component factor analysis, researchers investigating team process and performance in Navy teams identified four dimensions crucial to effective teamwork (refer to Table 1).

We have used these dimensions (1) to guide our exploration of where agent technology may provide value-added assistance, (2) in our implementation of agent architecture and agent interactions with the human team members in our demonstration scenarios and (3) to generate hypotheses and choose variables to manipulate and measure in our human experimental studies.

1. Information Exchange • Seeking information from all available sources • Passing information to the appropriate persons before being asked • Providing “big picture” situation updates	**2. Communication** • Using proper phraseology • Providing complete internal and external reports • Avoiding excess chatter • Ensuring communications are audible and ungarbled
3. Supporting Behavior • Correcting team errors • Providing and requesting backup or assistance when needed	**4. Team Initiative/Leadership** • Providing guidance or suggestions to team members • Stating clear team and individual priorities.

Table 1. NAWC/TSD ATOM teamwork dimensions

The ATOM model postulates that, besides their individual competence in domain specific tasks, team members in high performance teams must have domain independent team expertise, that is comprised of the different categories of Table 1. The performance of teams, especially in tightly coupled tasks, is believed to be highly dependent on these interpersonal skills.

In our research we investigated the role of agents as participants in human teams. The first step to investigating software agents as potential team members was to identify the characteristics human teammates would need to assess the capabilities and monitor the performance of these new teammates. This initial research was performed using simple human-agent dyads. In addition, we investigated the human agent interaction in situations where an agent is an assistant helping a particular team member perform a task, such as an agent that highlights high priority targets on a screen. The impacts of agent assistance of this sort were evaluated for both simple agents that aggregated and presented information and a more sophisticated route planning agent. Another possibility we investigated was to perform tasks for the team as a whole (many-to-one) such as monitoring communications to identify and record references to common targets. One of our experiments with a synthetic radar task compared this sort of team aiding with agent assistance at individual tasks. A third possibility was to operate as a fully autonomous team member, deciding what to do and explicitly coordinating with other team members to achieve team goals. This strategy was tested for a simple case for the synthetic radar task.

Besides the controlled laboratory experimentation, we explored the 3 types of assistance that agents can provide to human teams in a series of feasibility demonstrations that employ our RETSINA multiagent infrastructure software system in complex multiagent scenarios involving hybrid human-agent teams. Using teams of RETSINA agents we have created a wide variety of applications in areas ranging from financial portfolio management to air craft maintenance. A key advantage to such multiagent systems is their ability to access, assemble, and filter large amounts of digital information from diverse sources. For domains such as the modern battlefield in which massive amounts of information are already in digitized form agents could prove to be valuable teammates for humans who have difficulty in accessing and interpreting this electronic data

To aid human teams using teams of agents requires good models of teamwork for each. For agents the teamwork model must account for shared information such as goals or plans, models of themselves and other agents, and communications policies and protocols. For human teams the model must fit empirical data as well. The model we adopt for agent teamwork is based on theories of joint intentions (Cohen and Levesque 1990) and shared plans (Grosz and Kraus 1996). Joint intentions theory revolves around the problem of maintaining shared goals. Agents communicate to inform one another of their commitment to a goal (forming the team), goal achievement (more effort not needed), or decommitment (goal found to be unachievable). Shared plan theory extends coordination to a common high-level team model that allows agents to understand requirements for plans that might achieve a team goal. This allows team members to match their own capabilities to roles in the plans. Joint intentions theory solves problems such as multiagent foraging where team members can act relatively independently in pursuing the team's goal. Shared plans allow teammates to support one another directly. A robot with tracks, for example, might carry a wheeled fire-fighting robot across rubble to the location of a fire.

Much as traditional team research has grappled with issues surrounding coordination among human teams and the benefits of anticipating and interpreting situations (e.g., Entin & Serfaty, 1999; Fiore, Salas, & Cannon-Bowers, 2001; Foushee, 1984; Orasanu & Salas, 1993; Stout, Cannon-Bowers, Salas, & Milanovich, 1999), our models attempt to provide a framework through which agents are able to plan and coordinate around a shared task understanding. As such, for software agents to be effective teammates (for other agents or humans), there must be models of the teammate performance at differing levels of specificity. It is easy for an agent or robot to recognize its own impasses because it has direct access to its own goals and plans. To monitor another's performance, however, requires modelling the other's goals, plans, and actions with sufficient fidelity to differentiate between rescinded suspended, re-sequenced, and failed plans. Plan recognition at this level, especially recognizing the goals or plans of a human teammate, is an open problem. Because of all the difficulties inherent in imperfect plan recognition, our approach has been to use modelling as a basis for providing anticipatory assistance by agents to human team members without requiring precise plan recognition. In this form of agent-based team aiding, illustrated in the technological demonstrations, (section 3.1) the agents use models of the human team members' tasks to base intent inferencing so they can proactively seek, organize, and cache information likely to be needed in the near future.

Our technology demonstrations are intended to explore the limits to agent-based team aiding. The first of these demonstrations, *Jocasta*, showcased the abilities of RETSINA agents in a joint mission planning scenario to provide proactive assistance In the subsequent *Agent Storm* demonstration agents autonomously coordinated their team-oriented roles and actions while executing a mission in the *ModSAF* (Modular Semi-Automated Forces) simulation environment. The third demonstration, the Noncombatant Evacuation Operations (NEO) illustrates many of the potential uses of intelligent agents in human teams and the importance of developing these capabilities. Software agents were shown able to anticipate the information needs of their human team members, prepare and communicate task information, engage in planning and replanning in response to changes in situation or the capabilities of other team members, and effectively support team member mobility. In this chapter, we report in some detail on the agent-based team assistance in the Noncombatant Evacuation Operations (NEO) deomnstration.

3. Research Backdrop: Brief Overview of RETSINA

RETSINA provides a multiagent infrastructure for finding, assembling, and coordinating teams of agents to accomplish specified goals. We have developed RETSINA agents that support human teams, in supporting the task of individual team members, act as team members or support the team as a whole. In our implementations, we have been guided by the teamwork dimensions of the ATOM model, and the findings on high performance teams in the studies of Orasanu and Salas (1993) and Salas et al. (1992). In particular, the team aiding RETSINA agents:

(a) identify relevant recipients of critical information and forward the information to them taking into consideration user cognitive limitations
(b) keep track of task interdependencies among different team members
(c) recognize and report conflicts and constraint violations

(d) propose solutions to resolve conflicts
(e) monitor team performance and alert team members to problems.

RETSINA has been developed under the following assumptions: (a) the agent environment is open and unpredictable, i.e. agents may appear and disappear dynamically; (b) agents are developed for a variety of tasks by different developers that do not collaborate with one another, (c) agents are heterogeneous and could reside in different machines distributed across networks, and (d) agents can have partially replicated functionality and can incorporate models of tasks at different levels of decomposition and abstraction. For example, there can be a single agent that provides all kinds of weather information (including barometric pressure, wind direction etc) and for all cities in the world. On the other hand, there could also be weather agents that provide only temperature. Alternatively, there can be an agent that provides radar operator functionality, and agents that provide only target tracking functionality (a subtask of the radar operator task) for a particular environment. These agents could vary in fidelity to the task constraints (e.g. the target tracking agent could operate at a more refined resolution level for tracking).

To be an effective team member, besides doing its own task well, an agent must be able to receive tasks and goals from other (appropriate) team members (humans or agents), be able to communicate the results of its own problem solving activities to appropriate participants, monitor team activity and delegate tasks to other team members. A pre-requisite for an agent to perform effective task delegation is to (a) know which tasks and actions it can perform itself, (b) which of its own goals entail actions that can be performed by others, and (c) who can perform a given task. The individual agent architecture that we have developed (Sycara et al. 1996) includes abilities of agents to send messages to one another (RETSINA agents communicate using the KQML language), declarative representation of agent goals and planning mechanisms for fulfilling these goals. Therefore, an agent is aware of the objectives it can plan for and the tasks it can perform. In addition, the planning mechanism allows an agent to reason about actions that it cannot perform itself but which should be delegated to other agents. To do so, an agent needs ways to find out the capabilities of other team members, i.e. what tasks other agents can perform. As shown in Figure 1, each agent has a *communications* module (Shehory and Sycara 2000), which is responsible for interactions and the exchange of messages with other agents. These messages could contain new objectives from other agents or from the environment. The communicator uses the input/output message queue to modify the agent's set of high-level *objectives* in its knowledge store. The *planner* module (Paolucci et al. 2001) uses the objectives and a *plan library* of pre-specified plan fragments. The planner composes these plan fragments to construct alternative possible plans for the agent, stored as *task structures*. The *scheduler* module uses the task structures determined by the planner module to create a *schedule* of primitive actions for execution, that the agent can then execute. The *execution monitor* module monitors action execution in the operating environment and suggests repairs, if actions fail.

<*****Figure 1**: INSERT Individual RETSINA agent HERE ***>

The four modules operate in parallel, as multi-threaded code. Thus, the agent can receive messages from human team members or other agents through the communicator module while the planning module simultaneously constructs plans. In this way, an agent can interleave *deliberative planning* with *information gathering* and *execution monitoring*, an important capability in dynamically changing environments.

The green box in Figure 1 represents the knowledge store of the agent, which consists of a *goal stack*, where incoming or internally generated goals are stored and a *task database*, where task fragments relevant to the agent's functionality are stored and reused to construct plans. It contains an additional *belief database* that stores the current beliefs of the agent that can change due to evolving situation changes or due to agent-internal processing.

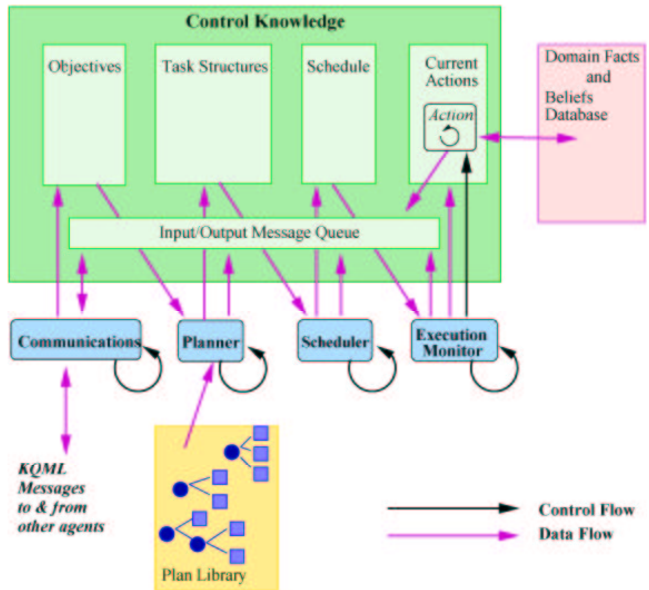

Figure 1 : Individual Retsina agent

Each RETSINA agent provides a set of services that are defined by the agent's capability/functionality. Figure 2 shows a typical multi-agent interaction, where a team of users specify goals and tasks to be accomplished. These specifications are received by *interface agents*, that decompose the task and hand off portions of the task to *task agents*. The task agents could accomplish the tasks themselves, with the help of other task agents or *information agents* (Decker et al. 1997a). If a task agent is unable to perform a task (due, for example to overloading), or the task is too complex, it may further decompose the task and delegate subtasks to other task agents. For example, an agent that plans possible routes for tanks, so they arrive at a rendezvous point at some particular time, may request the services of a weather agent for weather predictions in the area, since tanks can move at different speeds in wet soil than in dry soil. On the other hand, the route planner is also a service provider of route planning services that can be requested by a mission-planning agent.

Since every agent can both plan and execute action sequences, the above architecture enables *deliberation* and *reaction* to the environment to be performed as needed at every stage of the task decomposition. In addition, the system does not impose on agents a particular granularity of task decomposition. Based on the environment and on constraints passed down from other agents, a (task) agent can plan and choose the best course of action. This enables it to flexibly and dynamically adapt to changes in the environment including changes in the goals and intentions of its human and software teammates.

In addition to agents that request and provide services, RETSINA includes a category of agents called *middle agents* (Decker et al. 1997b, Wong and Sycara 2000). We have identified different types of middle agents, such as matchmakers, brokers, facilitators, recommenders, and have evaluated their performance tradeoffs experimentally (Decker et al. 1997b). We have developed and implemented an advertisement mechanism, i.e. an agent's capability self description that expresses the agent's commitment to perform the advertised task (Sycara et al. 1999, Wong and Sycara 2000). Capability descriptions enable humans and other machine agents to know which tasks to delegate to which agents and in addition it enables developing trust in the agent. Capability parameters could include cost, time to perform the service, precision of results, availability and reliability. Advertisements are sent to particular types of agents, called middle agents (Decker et al. 1997b, Sycara et al. 1999). When a human or agent needs to find an agent with a given capability in order to delegate a task to it, it sends a request to a middle agent. The middle agent matches the request to the set of agent capability advertisement in its data base. Knowing service parameters in addition to agent capabilities of a supplier agent allows a requester to (human or software agent) to make rational decisions about selecting a particular supplier to service a request, especially in environments with considerable uncertainty or changing decision criteria. For example, under situation X, a less precise but more timely response might be more desirable and vice versa in situation Y. Since in our system, agents are truthful and cooperative, their advertised parameters correspond to an agent's best knowledge of its current attributes. Middle agents allow the adaptive formation of teams on demand to fulfil requests and provide system extensibility, scalability and robustness.

3.1. Supporting Human Teams in Non-Combatant Evacuation Operations

RETSINA agents have been used to support teams in different military simulations including joint mission planning. In this chapter, we report on our experience in developing a demonstration of RETSINA agents supporting a human team in a Noncombatant Evacuation Operation (NEO) scenario. This demonstration illustrates the use of agent technology for cooperatively planning and executing a hypothetical evacuation of US civilians from a Middle Eastern city, let us call it Kabul, in an escalating terrorism crisis. In the scenario, agents are used to help the human team evaluate a crisis situation, form an evacuation plan, follow an evolving context, monitor activity, and dynamically re-plan.

The scenario unfolds as follows: The Year is 2010 and the location is a Kabul, Afghanistan. A conference is taking place there. Many of the conference participants are US citizens. A revolutionary group, called the Gazers, starts massive protests, including protests against the US and detonation of small incendiary bombs. The US Ambassador in Afghanistan forms a team with a representative of US Transportation Command (USTRANSCOM) in St. Louis, MI. and with the Joint Forces Commander (JFC) who is located in a US base in Germany. The three human team members must make decisions as the situation escalates. There are a number of software agents that support the human team. We describe the most important ones. First, there are 3 Voice Agents that are interface type agents. They receive the voice input from the human teams members, and through speech to text translation store the messages. Another type of interface agents, namely the Messenger agents, subsequently translate the human input into software-understandable code. This code indicates the goals of the human team members that the rest of the software agent system tries to plan for and fulfill. These human goals, of course, may change, as the situation evolves in the world. The Voice Agents and the Messengers are RETSINA agents and therefore posses planning and communication capabilities, as depicted in Figure 1. They communicate both with the human team members and also with one another through their RETSINA communication modules (see Figure 1). The Voice Agents and the Messenger agents aid the team as a whole since they:

- keep track of the communications of the human team members,
- keep track of the evolving goals of the human team members, new goals, which goals have been satisfied and which ones are still pending,
- keep track of which agent has been tasked to perform one or more particular tasks in fulfillment of different goals, and
- keep track of the current world state.

Besides the Voice and Messenger agents, there are a variety of other agents who play the role of teammates. There is the Webmate agent (Chen and Sycara 1998) that can search in the

Web to find multimedia information, including video clips and texttual articles that contain

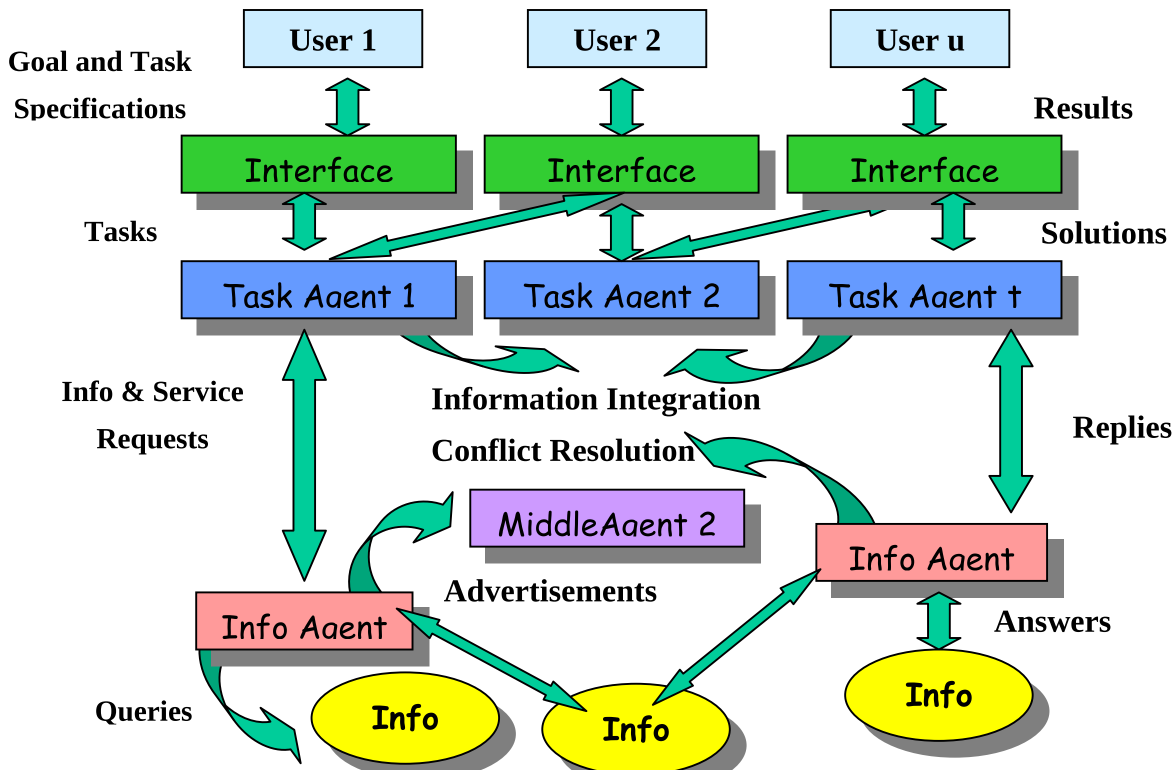

Figure 2. Retsina functional organization

information relevant to a particular input query; there are two Weather agents that can provide weather information for different parts of the world, there are two Airline Reservation agents, a Route Planning Agent that can plan routes taking into considerations constraints of non-traversibility of routes, a Visual Recognition Agent that can recognize events in the world, such as road blocks or explosions; there is CAMPS, a military asset deployment agent that can make schedules for the deployment of military aircraft. In addition, there are two kinds of middle agents, a matchmaker and a facilitator who facilitate the discovery and matching of agents that provide certain services (e.g. weather information) with agents that request these services (e.g. the Route Planning agent may want to find an agent that can provide weather information). A snapshot of these agents and their interactions through message passing (the dotted line in the display indicates real time message communication among the indicated agents) is shown via the Visualization agent in Figure 3.

As the human team members communicate, plan and collaborate, the agents that aid the team as a whole, namely the Voice Agents and the Messenger agents keep track of the human team members' communications and goals. They, then decompose these goals into subgoals that must be planned for and fulfilled. In the course of this decomposition, the Messenger agents through the services of the Matchmaker and Facilitator middle agents, find the appropriate agents to fulfill the current subgoals. For example, when the Ambassador hears that the terrorist group has exploded bombs in the city, he communicates to the JFC and the Air Operations officer that they should be starting to look for civilian transport for the 50 US citizens that are participating in the conference.

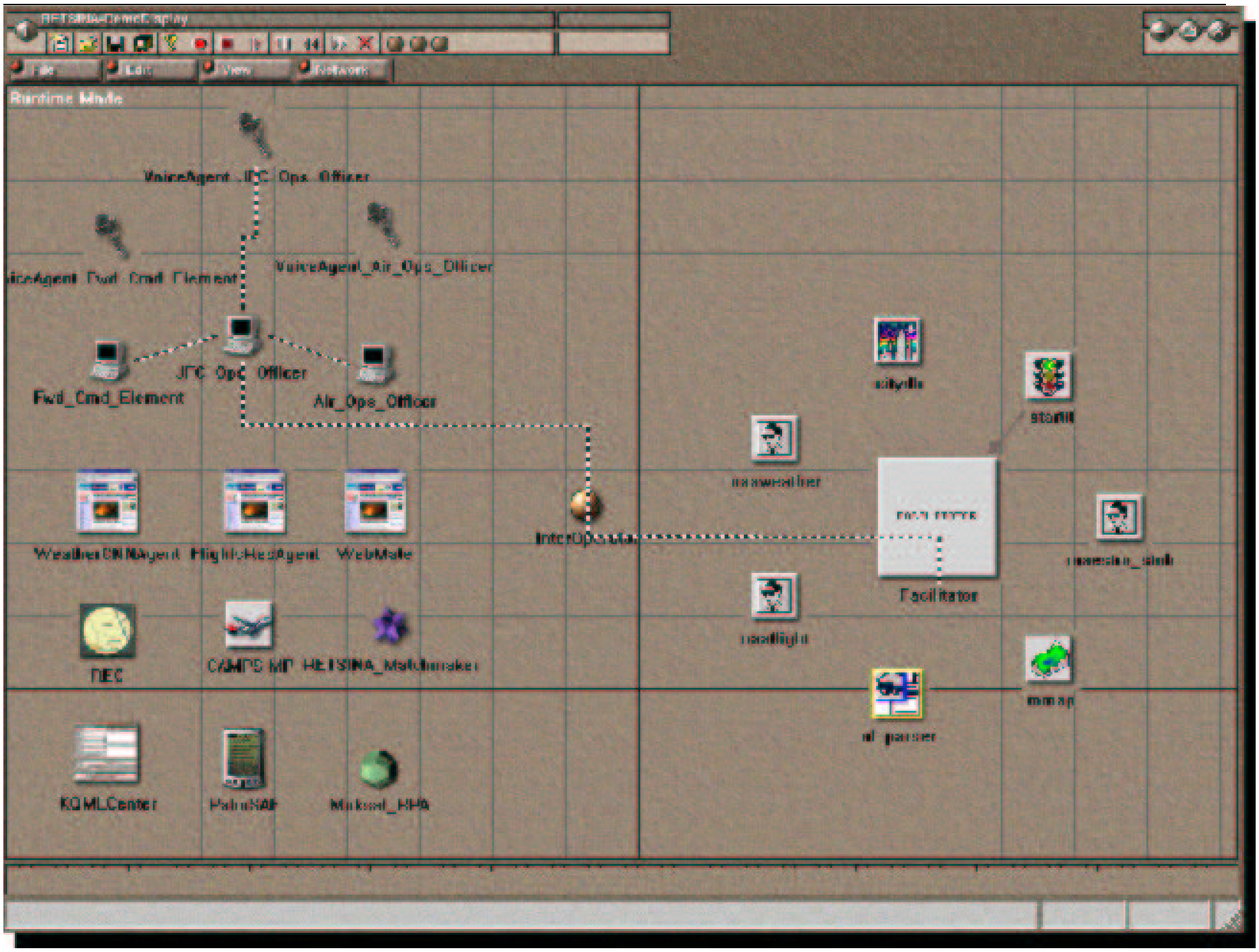

Figure 3. The Demo Display screen

The Ambassador's Messenger Agent understands this communication as a goal to be fulfilled, communicates this to the other

Messengers, and starts looking for agents that could fulfill this goal. An enquiry of the Matchmaker middle agent lets the Messenger know that there are two Airlines Reservation Agents that can fulfill the goal of getting schedules and availability of flights from Kabul to the US. The Messenger chooses one of the Airlines Reservation Agents, contacts it and enquires about availability of seats for flights to the US. The Air Reservation Agent returns a schedule of departing flights to the US. The Ambassador's Messenger Agent displays a message on the Ambassador's screen, saying that the flight schedule is ready for him to view whenever he wants to. In this way, the anticipatory assistance of the Messenger and Air Reservations agents is unobtrusive. In other words, the results of the information gathering are not imposed upon the Ambassador, with the possible risk of interrupting a highly critical and time stressed communication, but the Ambassador is simply notified so he can view the results whenever it is convenient for him. In addition, if the Ambassador wishes, he can share the flight schedules with the other human team members. To do this, the Ambassador, presses a button on his display, which notifies the Ambassador's Messenger Agent that it is wished for the display to be shared. The Ambassador's Messenger then communicates with the Messenger Agents of the Air Ops Officer and the JFC transmitting the flight schedules to them so they can notify the respective human team members and allow them to display the flight schedules on their respective screens.

The above example step in the NEO demonstration illustrates a number of innovations in agent-based team support as implemented in the RETSINA system. First, coordination of the Messenger Agents, the Matchmaker middle agent and the Air Reservation agent allows *anticipatory/proactive information* delivery. As soon as the Ambassador expresses a desire/goal that is relevant to the task of the human team, the appropriate agents are discovered and tasked without any further explicit tasking by the Ambassador. This ability implements in the agents the monitoring and cooperative behavior of humans that has been identified as one of the hallmark characteristics of human high performance teams. Second, RETSINA agents can also be invoked explicitly by a human team member, thus exhibiting information gathering and problem solving abilities. Third, it is not necessary for the Messenger Agent to know explicitly the name or location in the Infosphere of Air Reservation Agents; the Messenger can find such agents through enquiry to the middle agents about the desired agent functionality (yellow pages enquiry) rather than agent name. In a dynamic environment, where agents come and go unpredictably, it is not possible to know the exact name of an agent that provides particular services, hence service discovery by capability is the effective way to go. Fourth, after the Air Reservation Agent returns the flight schedules, the Messenger does not display them immediately on the Ambassador's browser, but rather it notifies the Ambassador as unobtrusively as possible, so the Ambassador can view the schedules at this convenience. This kind of human-agent collaboration is done for the following reasons: (a) the Ambassador may be engaged in other high priority activities that involve viewing of other results, hence he may not want to be interrupted; (b) the Ambassador's goal of finding airline flights for the evacuees may have changed during the time that it takes the Air Reservations Agent to find the flight schedule information; such goal changes may result from changes in the situation. Fifth, getting the flight schedules is coordinated by the Messenger of the human team member that initiated this goal, in our example the Ambassador. This is done so as to control the collaboration of the Messenger Agents and avoid redundant work with possibly confused results. Sixth, the results of the activity of the software agents are initially shown to the goal originator, in this example the Ambassador. It is up to the human goal originator to choose to share the results with the other members of the human team. This is done so as to control the information flows and display of the results for computational efficiency but also saving cognitive load for the human team members (in case it is not necessary for all human team members to view the results of agent computations). Figure 4 illustrates the interrelationships among the functions of the RETSINA agents, the cognitive processes of human team members and the team behaviors both support.

During the Non-Combatant Evacuation Demonstration, additional agent team aiding examples of interest include:

- When the need for a route to the airport is mentioned, the Route Planning Agent (RPA) is tasked by the Messenger Agent of the Ambassador. Given its access to a multi-modal map and knowledge of other contingencies, the RPA plans a route to the airport. It sends this plan to the Ambassador's Messenger, which then display a multi-modal map with a planned evacuation route.
- Through a Phone Agent, a site evacuation specialist in the field informs the system that the environment has changed: a roadblock interferes with the originally planned

route to the airport. This information is propagated from the VoiceAgent to the Ambassador's Messenger, which passes the message along to the RPA. The RPA returns a revised route to the Ambassador and subsequently to the rest of the human team members through their Messengers.

- The Visual Recognition Agent (VisRec) discerns that a bomb has exploded at the airport. It notifies all Messenger Agents. In this example the Visual Recognition agent provides situation assessment, situation monitoring and alerting of team members. All these abilities have been identified as characteristics of high performance human teams.
- Given the new information about the bomb explosion provided by the Visual Recognition agent's alert, the Joint Forces Commander recommends abandonment of the goal of evacuating the people through civilian flights and expresses the goal of deploying military airlift. The JFC's Messenger tasks the CAMPS Agent to create and validate a feasible military airlift schedule in cooperation with other agents that have information about the current location and availability of US military assets in the region. Meanwhile, the RPA is tasked to create a route for the US military airport in the host country that is closest to Kabul.

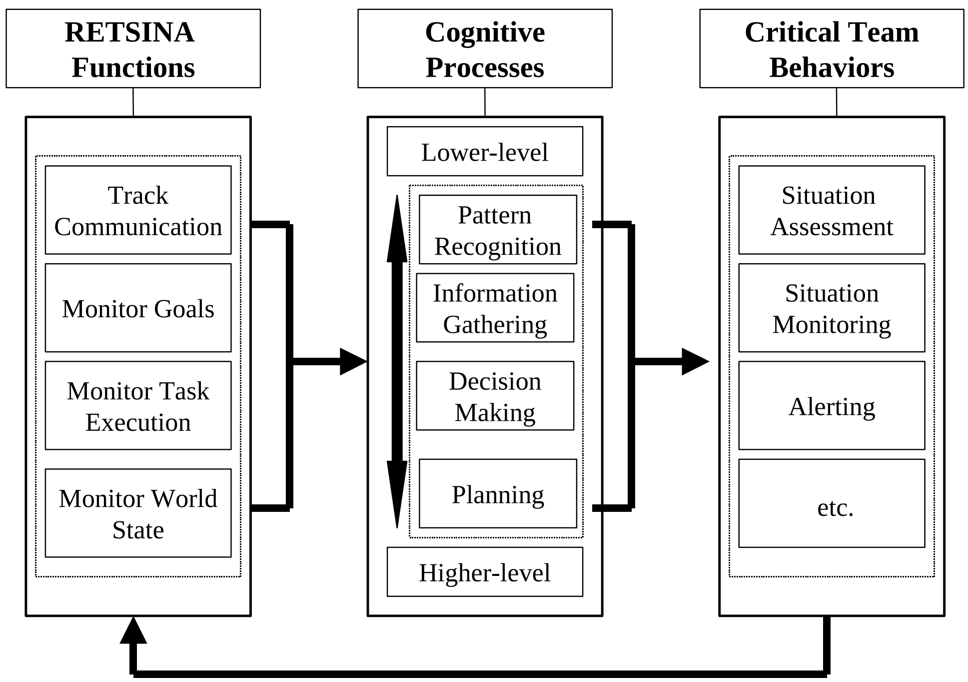

Figure 4. RETSINA functions and the cognitive processes they support

The above examples from the NEO system demonstration illustrate the types of human agent interaction and agent-based team aiding that the RETSINA system provides. These aiding strategies include:

1. *Aiding an individual human team* member in information gathering or planning tasks
2. *Acting as team members themselves*. In this capacity, RETSINA agents: (a) provide proactive and reactive information and planning support, (b) perform information gathering and planning tasks to promote the team goals, (c) perform task decomposition and task allocation to other members of the agent team so as to efficiently contribute to the team goals

3. *Aiding the human team as a whole*. In this capacity RETSINA agents (a) provide situation assessment, (b) monitoring and alerting team members to important situation changes, (c) communicating their results in unambiguous, multimodal and non-intrusive ways, (d) discover (through middle agents) suitable additional team members and information sources that can aid the team.

Unlike most examples of human teams where the team members are statically known a priori, RETSINA does not make any such closed world assumptions but *allows dropping, adding, and discovering new teammates dynamically*. This functionality reflects the requirements of real situations (especially military situations where teammates may become incapacitated and others must be found to take up their roles).

4. Experiments in Agent Supported Human Teamwork: Target Identification Experiments

The RETSINA technology demonstrations have substantiated the feasibility of deploying networks of intelligent agents capable of individual initiative and coordination as well as tasking by humans. The very complexity and sophistication of these demonstrations, however, make it difficult to evaluate basic factors affecting human ability and willingness to work with agent teammates. In a parallel series of experiments with human subjects we have investigated the factors theories of teamwork suggest might affect our ability to work with software agents.

In the first experiment, teamwork was investigated in its simplest form using human-agent dyads to provide a controlled setting to explore the roles of intelligibility and trust. In a second experiment, agents' tasks were again simplified to aggregation of data and record keeping in order to examine the relative effectiveness of aiding "domain independent" teamwork tasks. In the third series of experiments, agents were limited to supporting individuals but called upon to help in a more complex deliberative team planning task.

4.1 The Tandem Simulation Environment

The first two experiments used a moderate fidelity simulation (TANDEM) of a target identification task, jointly developed at the Naval Air Warfare Center-Training Systems Division and the University of Central Florida and modified for these experiments. The TANDEM simulation was developed under the TADMUS (tactical decision making under stress) program of the US Office of Naval Research and simulates cognitive characteristics of tasks performed in the command information center (CIC) of an Aegis missile cruiser. Figure 5a. shows a typical TANDEM display. Information about the hooked target (highlighted asterisk) is obtained from the pull-down menus 'A','B', and 'C'.

The cognitive aspects of the Aegis command and control tasks which are captured include time stress, memory loading, data aggregation for decision making and the need to rely on and cooperate with other team members (team mode) to successfully perform the task. The more highly skilled tasks of the individual team members that involved extracting and interpreting information from radar, sonar, and intelligence displays is not modeled in the simulation. Instead of interpreting displayed signals to acquire diagnostic information about targets, TANDEM participants access this information manually from menus. In accessing new information, old information is cleared from the display creating the memory load of simultaneously maintaining up to 5 parameter values and their interpretation.

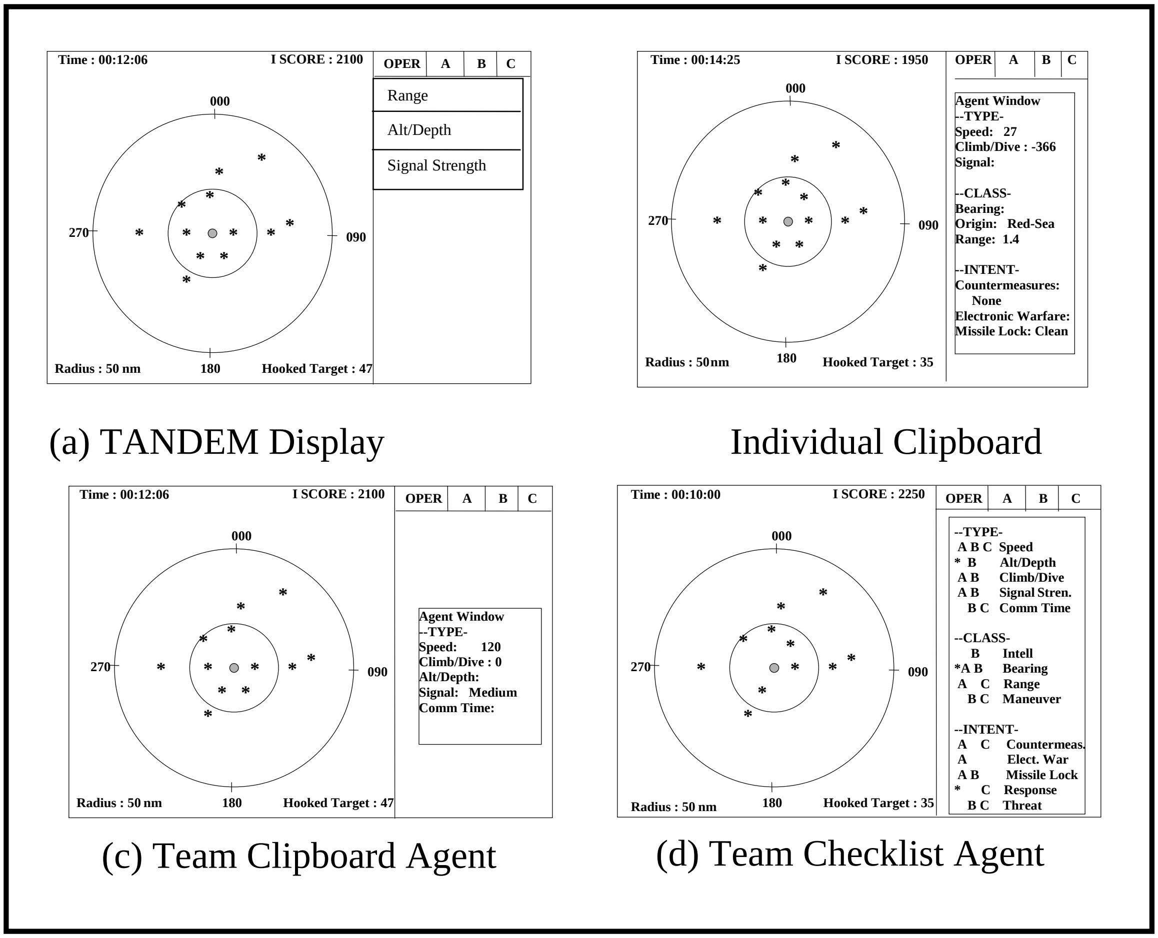

Figure 5. TANDEM components with and without agent aiding

In the TANDEM task subjects must identify and take action on a large number of targets (high workload) and are awarded points for correctly identifying the targets (type, intent, and threat). and taking the correct action (clear or shoot). A maximum of 100 points is awarded per target for correct identification and correct action. Users “hook” a target on their screen by left-clicking on the target or selecting “hook” from a menu and specifying a target’s unique contact number. Only after a target is hooked can they access information relative to that target. In team configuration TANDEM consists of three networked pc’s each providing access through menus to five parameters relative to a “hooked” target. Their tasks involve identifying the type of contact (submarine, surface, or aircraft), its classification (military or civilian), and its intent (peaceful or hostile). Each of these decisions is made at a different control station and depends on five distinct parameter values, only three of which are available at that station. Subjects therefore must communicate among themselves to exchange parameter values to classify the target. If the team finds a target to be hostile it is shot, otherwise it is cleared and the team moves on to another target.

In standalone mode all of the information is made available on a single pc with the station specific parameters accessed using three distinct menus. Menus in standalone mode present 5 parameters each. In team mode the three menus present 3 (overlapping among team members) parameters per menu. Just as TANDEM simulates cognitive aspects of the Aegis missile command and control task, it provides a context to simulate the gathering, aggregation, and presentation of communications, command, control and intelligence information by intelligent agents. To investigate impacts on human-human coordination presentations of aggregated information can be tailored to support different aspects of the participants' cognitive tasks.

4.2. Trust, Error, and Uncertainty

Many of the complex of issues involving mutual human-machine modeling, awareness, and coordination are captured by the anthropomorphic term "trust". If we examine the considerations that enter into our decision to delegate a task to a subordinate, instruct the subordinate in how to perform the task, monitor that performance, or authorize some class of tasks without follow-up, our trust in the subordinate will almost certainly play an explanatory role. Closer consideration will show our use of the term to be multidimensional. The trust we have that our secretary will remember to pick up the mail, is distinct from our trust that she will compose a postable business letter, which in turn is distinct from our trust in the lawyer who assures us that the letter is not actionable. A merger of several taxonomies proposed by Lee and Moray (1992) distinguishes:.

1) trust which is based on observed consistency of behavior (*persistence* or *predictability*) I trust my watch to keep relatively accurate time

2) trust which is based on a belief in competence or well formedness (*competence* or *dependability*) I trust the recipe for hollandaise

3) trust which is based on faith in purpose or obligation (*fiduciary responsibility* or *faith*) I trust my physician to monitor my health

As bases for human modeling of machine agents this taxonomy suggests that agents could be made predictable by: 1) consistently pairing simple observable actions with inputs, or 2) making the causes and rules governing an agent's behavior accessible to the human or 3) making the purpose, capability, and reliability of the agent known to the user. Muir (1994) refers to the process of acquiring predictive models of these sorts as *trust calibration*. The idea is that performance will be better for human-machine systems in which trust is accurately calibrated. Such precision allows the human to make more accurate predictions. The greater predictability of *consistent* or *competent* agents should also make boundary conditions and limitations more apparent and remediable. Agents trusted on faith, by contrast, might require a higher degree of reliability across their range and more communication to maintain accurate coordination. Research in this area (Muir 1994; Muir and Moray 1996; Lee and Moray 1994; Lewandowsky, Mundy, and Tan 2000) has focused on the relation between trust and users' reliance on automation prior, and subsequent to controller failures at process control tasks. Moray, Inagaki, and Itoh (2000) recently extended this approach to decision making for automated diagnosis and fault management.

In the TANDEM environment we investigated the roles a very simple information agent might play and the impact of errors that undermined the presumed bases of trust for each of these roles. The experiment addresses the questions of task allocation (under what conditions should automated information processing be curtailed or eliminated) and information presentation (can choice of presentation context affect the usability of processed information). We hypothesized that effective human/agent performance might require precise calibration of trust so that the decision maker could accurately interpret the agent's communications and anticipate its limitations.

4.3. First TANDEM Experiment: human-agent dyads

We constructed information presentations of three types, which roughly parallel the level of trust they rely upon for interpretation:(1) aggregated (*list*), (2) integrated (*table*), and (3) synthesized (*oracle*). The reported experiment pairs error-making and error-free data presentations to observe the effects on decision quality, reliance on agent provided information and reported confidence. .

4.3.1. Displays

Each agent provided one of three possible information presentations. To manipulate the subjects' trust, errors were introduced into the agents' presentations although the menus continued to provide correct values. In the control conditions agents presented errorless values as well. Possible errors were of three types: data errors (display levels 1,2,&3), classification errors (display levels 2&3), or decision errors (display level 3). Data errors occurred when the *list* agent displayed different data than the ground truth. For the other displays, erroneous input(s) were treated as legitimate leading to false but correctly classified entries in the table and effects ranging from inaccurate certainty factors to incorrect decisions (multiple errors) for the *oracle*. This type of error was explained to subjects as "problems with the agent's sensors". In classification errors values were reported correctly but were misclassified by the agent. Classification errors occurred when the *table* agent placed data in the wrong column(s). To *oracle* users classification errors appeared similar to data errors affecting the reported certainty factor or the decision for the right combination of errors.

Decision errors occurred only in level 3 when the *oracle* agent assigned an incorrect "type" to a target. Unlike incorrect decisions induced by data or classification errors, these "decision errors" were not accompanied by low certainty factors. Classification and Decision errors were explained to the subjects as "software problems". The problem of verifying and correcting errors varies with error type and agent presentation. Data errors only required comparing presented parameter values with those found on the menus. Classification errors required comparison between the displayed category of a parameter and the proper categorization of its value (*table*) or seeking additional information for low certainty factors (*oracle*). Detecting decision errors (*oracle*) required performing the classification task manually and comparing the results.

Errors of the different types were equated by matching corresponding rates to the distribution followed by data errors that were independent with probability of .33 per parameter. So, on an *oracle*'s display, for example, data errors, classification errors, and decision errors would each produce the same rate of erroneous "decisions" by the agent. Decision errors were therefore more rare but less easy to detect than the other forms.

Only one type of error was presented during a TANDEM session. Button presses to access agent information, menu selections, target hooks, classifications, and final actions and times were collected along with simulation states for each subject. Agent displays tested showed:

1) aggregated information (*list*) -- a list of parameters and values

2) integrated information (*table*) -- a table showing categorized values

3) synthesized information (*oracle*) -- target type assignment with certainty factor.

4.3.2. Method

Participants

Eighty-two paid participants recruited from the University of Pittsburgh community took part in the experiment. Reported results are based on complete data from 78 participants and partially complete data from an additional two.

Materials

Sixty targets were distributed in several concentric rings on the screen. The circle closest to the center is referred to as the "circle of fear" and the amount of time a target spent in this circle before being identified was measured as penalty time. The number of targets identified while in this penalty circle, targets identified outside of the penalty circle, and targets hooked but not resolved were all measured. Ratings of "trust" of simulated information agents using scales developed and validated by Muir (1994) were also gathered from each participant. These ratings, on a scale of 1(low) to 5 (high) focused on issues of dependability, predictability, accuracy, reliability and an overall "assessment of trust in the agent.

Procedure

Participants received standard instructions and a sheet of tables showing the correspondence between parameter values and identification decisions. Subjects were assisted through a five minute training session operating the simulation and then completed two 15 minute experimental trials, concluding the session by completing a "trust in automation" (Muir 1994) survey.

4.3.3. Results

Performance was analyzed using a repeated measures analysis of variance with session as the within subject factor and types of error and level of agent as between group factors. Effects of session were significant ($p < .05$) for each of the dependent measures reported. Where differences were found between groups, data were pooled across the two sessions and Post hoc analyses conducted using Fisher's LSD to identify reliable differences among the conditions.

Productivity

An ANOVA on the number of targets engaged within the penalty circle showed a main effect of type of error $F(3,71)=5.687$, $p=.002$ (No error mean=9.71, SD=5.28; Data error mean=6.98, SD=4.43; Classification error mean=9.03, SD=5.33; Decision error mean=11.63, SD 4.43). Post hoc testing showed that the No error ($p=.004$) and the Decision error ($p=.001$) means differed from the data error mean. Similar effects were noted for other productivity measures with data errors impairing performance more than others.

Accuracy

Measures of accuracy such as the percentage of correct responses to the aided air/surface/sub decision were affected by both agent role,

An ANOVA on correct responses to the aided air/surface/sub decision showed main effects for agent role, $F(2,69)=3.215$, $p=.046$ (List mean=.9165, SD=.1230; Table mean=.9027, SD=.1564; Oracle mean=.9614, SD=.08143) and type of error, $F(3,69)=2.924$, $p=.04$ (No error mean=.9569, SD=.0954; Data error mean=.8858, SD=.1546; Classification error mean=.9494, SD=.1000; Decision error mean=.9413, SD=.0913). In post hoc tests a difference was found between Oracle and Table means ($p=.021$) while the differences for No error ($p=.007$) and Classification error ($p=.042$) means with respect to Data errors again showed Data errors to be more damaging.

Accuracy was improved even for the unaided civilian/military decision as shown by the main effect for agent role on the unaided decision $F(2,61) =3.539$, $p=.035$ (List mean=.8049, SD=.2528; Table mean=.8675, SD=.2099; Oracle mean=.9310, SD=.1556).

Reliance on Agent

An ANOVA on use of the agent found main effects for both agent role, $F(2,71)=5.83$, $p=.005$ (List mean=7.45, SD=12.49; Table mean=9.21, SD=11.70; Oracle mean=16.19) and type of error, $F(2,71)=3.693$, $p=.016$ (No error mean=16.22, SD=16.16; Data error mean=7.95, SD=10.36; Classification error mean=8.88, SD=10.66; Decision error mean=15.06, SD=8.18). Post hoc tests found that the Oracle mean differed from both the List ($p=.007$) and Table ($p=.021$) means showing that not only did Oracle users perform better, they made greater use of their agent. The No error mean differed from the Data error mean ($p=.006$) and approached significance ($p=.051$) with respect to the Classification error mean showing again the harmful effects of data errors and the relatively benign effects of Decision errors. Subjects' ratings on 10 of the 11 scales of Muir's trust in automation questionnaire were lower for agents committing errors ($p < .05$) confirming the relationship between subjective ratings of trust and actual use of automation.

The hypothesized interaction between agent role and associated error type was not observed in these data. Instead, we found a clear pattern in which data errors were most harmful to accuracy and productivity while classification errors were least harmful. The *Oracle* led to better performance on both the aided decision (air/surface/sub classification) and unaided decisions (civilian/military classification). Not only did participants access the Oracle more often, but the Oracle appeared more robust under error conditions than either of the other agents. Figures 6 and 7 showing performance for subjects using the three agents under no error and data error conditions illustrates this advantage.

Our concern that errors that undermined the basis of trust required by an agent would be particularly detrimental was disconfirmed for the tasks we examined. The cognitive efficiencies offered by the *Oracle* appear to outweigh any difficulties of verification. Regardless of their source, errors affected participants' performance, reliance on agents and ratings of trust in a similar manner. Contrary to our expectations, agent intelligibility did not appear to affect the subjects' ratings of trust or productivity.

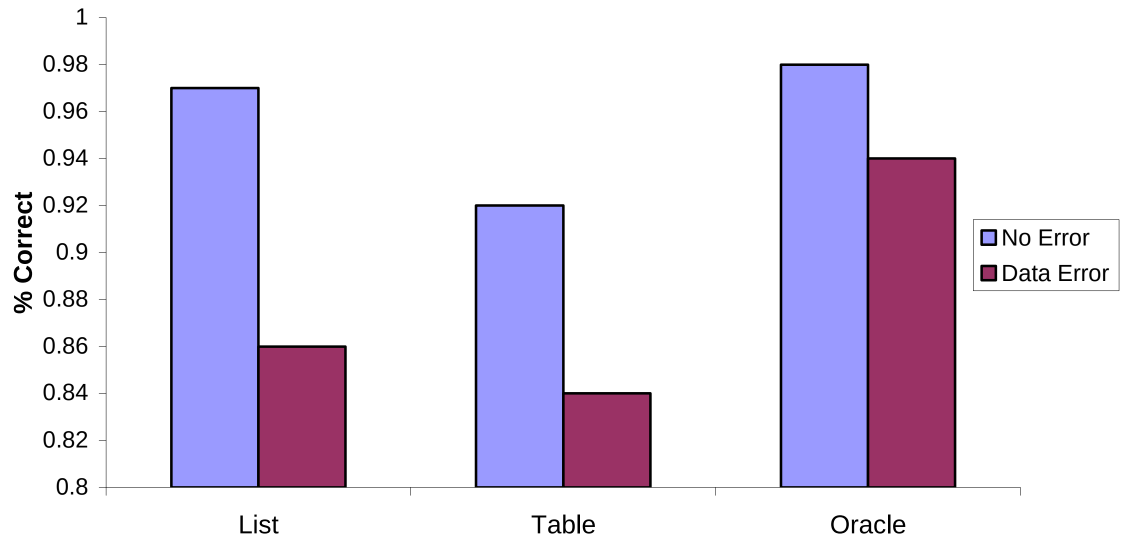

Figure 6. Accuracy for the aided (air/surface/sub) decision

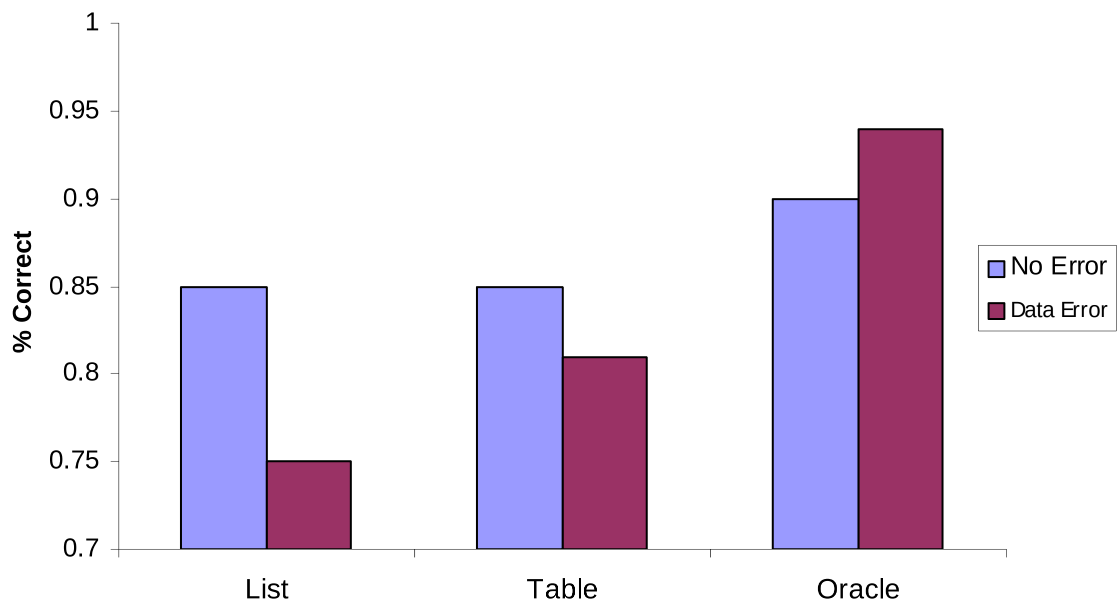

Figure 7. Accuracy for the unaided civilian/military decision

In a more detailed examination of decision strategies for error conditions we have compared participant accuracy for the aided decision with that of models for each condition matched on the number of menu accesses retrieving decision related values. This comparison measures the efficiency with which participants sought and used verifying information. Although Oracle users accessed menu items at roughly the same rate as List users their accuracy approached that of their model and beat the baseline strategy of always accepting the agent's decision. In the other two conditions accuracy is closer to that predicted by a baseline model using only information available through the agent (approximately 80%).

These results are encouraging for incorporating fallible agents into human teams because they suggest that the intelligibility of an agent's behavior (trust levels 1-consistency and 2- competence) may be less important than simply finding useful identifiable functions (trust level 3-purpose) when task demands are high

4.4. Second Tandem Experiment: Supporting Individuals vs. Teams

The second team TANDEM study examines different ways of deploying machine agents to support multi-person teams: 1) supporting the individual (within a team context) by keeping track of the information he has collected and helping the individual with his task and with passing information to teammates (Individual Clipboard, Figure 5b); 2) supporting communication among team members by automatically passing information to the relevant person which should reduce communication errors and facilitate individual classification (Team Clipboard, Figure 5c); and 3) supporting task prioritization and coordination by providing a shared checklist of which team member had access to which data (Team Checklist, Figure 5d). We hypothesized that the Individual Agent should aid the individual task and aid communication among team members. This agent shows all data items available to an individual team member (in this case, ALPHA) and fills in the values for the data items as the subject selects them from them from the menu. The values under the TYPE heading assist the individual with his task while the other team members may need to request the remaining values. The Team Clipboard Agent should also aid the individual task and aid team communication to a greater degree than the Individual Agent. This agent aggregates values from all members of the team to help the individual with his/her task. It automatically passes values as they are selected from a menu to the appropriate team member. Thus, when altitude/depth is selected from some one else's menu, it is passed to an individual team member (ALPHA) who can use it to make the type identification. We hypothesized that this agent should reduce verbal communication among team members and reduce communication errors. The third agent, Team Checklist, should aid team coordination. This agent shows who has access to what data. For example, all three team members (ALPHA, BRAVO, CHARLIE) have access to speed, but only BRAVO has access to "Intelligence". The final condition is a control where we observed team performance without the aid of any machine agent. This is the standard TANDEM task described in Smith-Jentsch, et al. (1998b). The goal of the study is to examine the impact of the aiding alternatives on: 1) communication patterns, 2) data gathering strategies, 3) reliance (i.e., use of) on the agents, and 4) performance.

4.4.1. Method

Teams of three subjects were recruited for this study. Each team was assigned to one of four conditions: 1) control, 2) individual agent, 3) team clipboard agent, or 4) team checklist agent. TANDEM was used with three-person teams, each member with a different identification task to perform (air/surface/submarine, military/civilian, and peaceful/hostile). One person was assigned to ALPHA, one to BRAVO and one to CHARLIE. ALPHA, BRAVO and CHARLIE had different items on their menus and different tasks during the trials. ALPHA identified the type of target (air, surface or submarine); BRAVO determined whether the target was civilian or military; CHARLIE determined whether the target was peaceful or hostile. In addition, CHARLIE acted as the leader by indicating the type, classification and intent of each target to the system and taking the final action (shoot or clear).

There were five pieces of information for each identification task, three of which must agree in order to make a positive identification. These pieces of information were distributed among the three team members. Each team member saw different data items on the menus and had three data items required for his/her identification task and several other items that the other team members might need to complete their tasks. Thus, the subjects needed to communicate with one another to perform their tasks for roughly two-thirds of the targets. All five pieces of information might agree for a particular target, however, in many cases, the ambiguity of the data was manipulated such that only three pieces agreed.

Materials

Targets were divided into three groups: 1) easy—all three pertinent items on the individual's menu agree; 2) medium—only two items on the menu agree, a team member must ask one or both teammates for data; and 3) hard—two items on the menu agree, but do not provide the correct solution. For example, ALPHA's task was to identify the type of target. If the target was easy, all three items on ALPHA's menu indicated the same type (e.g., air). If the target was of medium difficulty, one or two values would indicate air and the other might indicate submarine. If the target was hard, two of ALPHA's menu items might indicate air, but the remaining three items one from ALPHA's menu and two from the other team members' menus might indicate surface. Thus, the target would be a surface vessel. Subjects had no way of knowing the difficulty level of the targets.

Participants

Forty teams of three were recruited for this study (10 teams in each of the four conditions). Participants were recruited as intact teams, consisting of friends or acquaintances. Six teams were eventually dropped due to problems with data collection and one team in the Individual Checklist condition was dropped due to poor (29%) accuracy in target identification.

Method

Each team participated in a 90-minute sessions which began with a 15-minute training session in which the TANDEM software and team goals were explained. The team was told to identify as many targets as possible, as accurately as possible during the 15-minute trial. After the training session, the team participated in three 15-minute trials. At the conclusion, subjects were asked to complete a brief questionnaire.

Several forms of data were collected during the trials: 1) performance data from Tandem logs including the type and number of targets hooked and classified, the percentage of targets correctly identified, and the number of times the agents were consulted; 2) communication data encoded from observers or audio tapes including the number of requests for data (e.g., does anyone have initial range?), the number of responses (e.g., range is 5.6 nm), the number of target identifications (e.g., it's civilian), and the number of confirmations (e.g., target is sub, civilian); 3) observer data including ratings on team communication, situation assessment, leadership ad supporting behaviors; and 4) questionnaires completed by the subjects before they left.

4.4.2. Results

An ANOVA on the number of correctly processed targets showed a main effect for type of agent, $F(3,29)=3.961$, $p=.018$ (Control mean=10.29, SD=5.4; Individual clipboard mean=12.52, SD=5.88; Team checklist mean=15.96, SD=5.26; Team clipboard mean=16.67, SD=6.06). A similar effect is noted for correctly processed hard targets, $F(3,29)=4.518$, $p=.01$ (Control mean=2.83, SD=2.08; Individual clipboard mean=3.67, SD=2.57; Team checklist mean=5.04, SD=2.33; Team clipboard mean=5.29, SD=2.33). For hard targets post hoc tests found both of the team aiding displays (Team checklist, $p=.008$; Team clipboard, $p=.004$) differed from the control condition and the Team clipboard was found to differ from the Individual clipboard ($p=.04$) as well. This ranking is illustrated in Figure 8 with the Individual Clipboard falling between the control and team conditions.

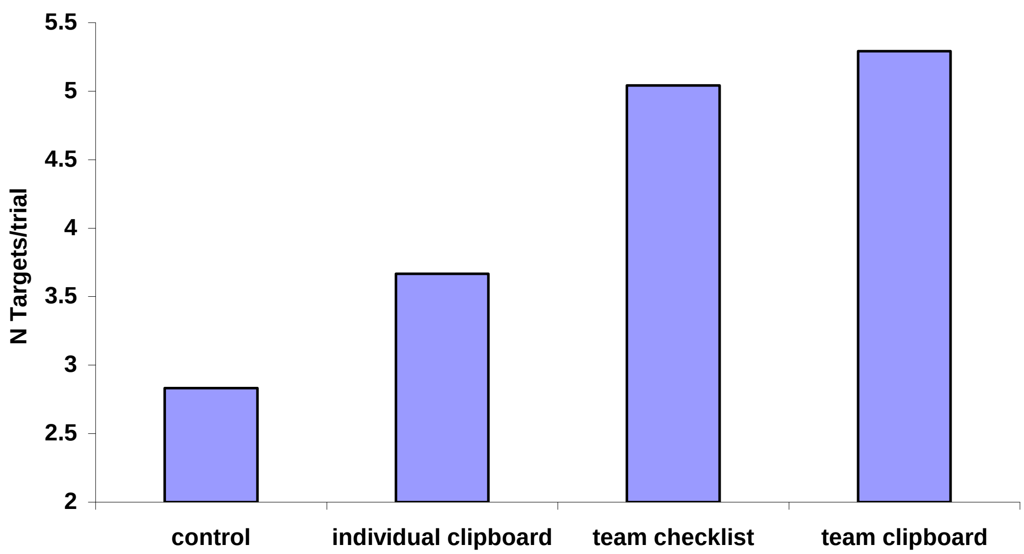

Figure 8. Aiding teamwork led to better performance on hard targets

In this study aiding teamwork directly (team clipboard/checklist) appeared more effective than supporting team members at their individual tasks despite the reductions in memory load and ready accessibility to parameters for sharing provided by the individual clipboard. The potential for coordinating human-human interactions through agent systems seems a particularly promising approach because of the high payoff and the reusable and largely domain independent character of the team supporting tasks.

The TANDEM experiments have demonstrated the value of agents for fast paced high workload tasks. Not surprisingly, we found that agent aiding was most effective when it addressed the most demanding aspects of the teams' task. In the human-agent dyad experiment this crucial factor was the memory load imposed by simultaneously judging evidence for three decisions. Under these conditions the performance of the dyad was highest when the agent assumed full responsibility for the classification of target type freeing the human to devote resources to the other two decisions. Sharing responsibility in this way resulted in greater accuracy for both the agent's and the human's decisions even when the agent made errors. In the second team experiment efficiently requesting and exchanging data were the crucial, resource consuming activities. The agents which supported these activities directly were more effective than the *clipboard* agent which acted to reduce memory load, a less critical factor in this experiment. The use of agents to support teamwork activities such as information exchange, communication, and monitoring appears especially promising because these activities are crucial across a variety of domains and might be addressed through development of a common agent infrastructure and interaction techniques.

Our experiments suggest that trust, fallibility, and intelligibility may not always need to be major concerns in introducing software agents into human teams providing their help can contribute to task performance in an evident way. As other studies of trust in automation (Muir 1994; Lewandowsky, Mundy, and Tan 2000, Moray, Inagaki, and Itoh, 2000) have found, our participants reacted to agent errors by decreasing their use of the agents and rating them as less trustworthy. The minor effect of Decision errors on *Oracle* users runs counter to both our original hypothesis about undermining trust and findings such as Lee and Moray's (1992) that the influence of errors on trust was closely related to consequence. In our experiment conditions are precisely matched on consequence so that users relying solely on information provided by the agent would make the same errors in classifying targets. The observable error rates differ substantially, however, with participants in the List and Table conditions seeing the errors as they occur while the Oracle users may become aware of errors only when they coincide to produce a wrong decision. While this rarity of overt errors does not quite raise the *Oracle* above the 90% reliability threshold Moray, Inagaki, and Itoh (2000) hypothesize may be necessary to blunt the effects of unreliability it appears to have done so in our study. For developers of information agents particularly those interrogating multiple sources this is good news because it suggests that apparent increased reliability of data summarization may make it more attractive as well as more useful to human analysts.

5. Experiments in Agent Supported Human Teamwork: Using the Infosphere to make Plans

Typically, human decision-makers, particularly military commanders, face time pressures and an environment where changes may occur in the task, division of labor, and allocation of resources. Information such as terrain characteristics, location and capabilities of enemy forces, direct objectives and doctrinal constraints must all play a part in the commander's decisions. Software agents have privileged access to the masses of information in the digital infosphere and can plan, criticize, and predict consequences from this sea of information with greater accuracy and finer granularity than a human commander could. Information within this infosphere can be used for data fusion, "what-if" simulations, or visualized to provide situation awareness. There is also, however, information that may not be explicitly represented electronically and is therefore inaccessible to software agents. Such information includes intangible or multiple objectives involving morale, the political impact of actions (or inaction), intangible constraints, and the symbolic importance of different actions or objectives. Before agents can consider information that is outside their infosphere, this information must be re-expressed in agent-accessible terms.. Military commanders, like other professional decision-makers, have vast experiential information that is not easily quantifiable. Commanders must deal with idiosyncratic and situation-specific factors such as non-quantified information, complex or vaguely specified mission objectives and dynamically changing situations (e.g., incomplete/changing/ new information, obstacles, and enemy actions). In order to cooperate with software agents in planning tasks commanders must find ways to translate these intangible constraints into tangible ones their agents can understand. The issue therefore becomes how should software agents interact with human teams to assist with problems which may be vague, ill-specified, with multi-attribute goals.

5.1. *MokSAF:* A Team Planning Environment:

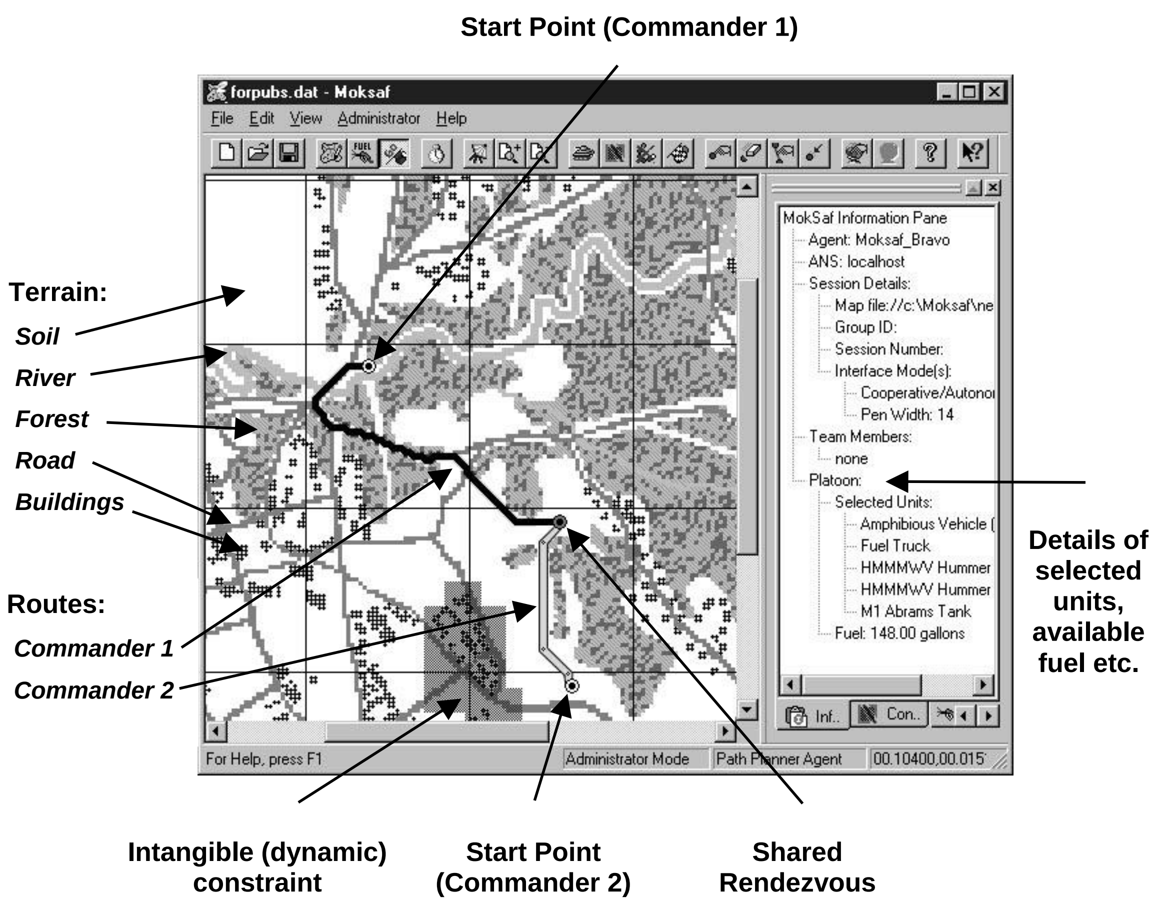

Figure 9. MokSAF display

We have developed a computer-based simulation called *MokSAF* to evaluate how humans can interact and obtain assistance from agents within a team environment. *MokSAF* is a simplified version of a virtual battlefield simulation called ModSAF (modular semi-automated forces). *MokSAF* allows two or more commanders to interact with one another to plan routes over a particular terrain. Each commander is tasked with planning a route from a starting point to a rendezvous point by a certain time. The individual commanders must then evaluate their plans from a team perspective and iteratively modify these plans until an acceptable team solution is developed.

One of the interface agents used within the *MokSAF* Environment is illustrated in Figure 9. This agent presents a terrain map, a toolbar, and details of the team plan. The terrains displayed on the map include soil (plain areas), roads (solid lines), freeways (thicker lines), buildings (black dots), rivers and forests. The rendezvous point is represented as a red circle and the start point as a yellow circle on the terrain map. As participants create routes with the help of a *route-planning agent* (see below), the routes are shown in bright green. The second route shown is from another *MokSAF* commander who has agreed to share his planned route. The partially transparent rectangles represent intangible constraints that the user has drawn on the terrain map. These indicate which areas should be avoided when determining a route.

5.2. Route-Planning Agents

Three different *route-planning agents* (*RPA*) have been developed to interact with the human team members in the planning task. The first agent, the *Autonomous RPA*, performs much of the task itself. This agent acts like a "black box." The agent creates the route using its knowledge of the physical terrain and an artificial intelligence planning algorithm that seeks to find the shortest path. The agent is only aware of physical constraints, which are defined by the terrain map and the platoon composition, and intangible constraints, which are graphically specified by the commanders.

The second agent, the *Cooperative RPA*, analyzes routes through a corridor drawn by the human team members, selects the optimal route and helps them to refine their plans. In this mode, the human and agent work jointly to solve the problem (e.g. plan a route to a rendezvous point). The workload should be distributed such that each component matched to its strengths. Thus, the commander, who has a privileged understanding of the intangible constraints and utilities associated with the mission, can direct the route around these constraints as desired. However, the commander may not have detailed knowledge about the terrain, and so the agent can indicate where the path is sub-optimal due to violations of local physical constraints such as traversing swamp or wooded areas.

The third condition, the *Naïve RPA* (or control), provides minimal assistance to the human commanders in their task of drawing and refining routes. Using this RPA, the commander draws a route that the agent then critiques for constraint violations such as impassible terrain or insufficient fuel. The commander is allowed to iteratively alter his failed route until a plan is found which passes muster. All three RPAs are intended to be used for iterative cooperative refinement of routes and the task of coordinating with other commanders requires continuous replanning as the team searches for its own best solution..

5.3. Experimental Methodology

The *MokSAF* experiments examine a deliberative, iterative and flexible planning task. There are three commanders (Alpha, Bravo and Charlie), each with a different starting point but the same rendezvous point. Each commander selects units for his/her platoon from a list of available units. This list currently contains M60A3 tanks, M109A2 artillery units, M1 Abrams tanks, AAV-7 amphibious assault vehicles, HMMWVs (i.e., hummers), ambulances, combat engineer units, fuel trucks and dismounted infantry. This list can be easily modified to add or delete unit types. With the help of an RPA, each commander plans a route from his starting point to the rendezvous point for the specified forces.

Once a commander is satisfied with the individual plan, she can share it with the other commanders and resolve any conflicts. Conflicts could arise due to several issues including shared routes and/or resources or the inability of a commander to reach the rendezvous point at the specified time. The commanders also must coordinate regarding the number and types of vehicles they can take to the rendezvous because their mission specifies the number and composition of forces needed at the rendezvous point. Commanders were additionally instructed not to plan routes that took them on the same paths as any other commander which required them to coordinate routes to avoid shared paths.

5.3.1. Participants

Twenty five teams consisting of three-persons were recruited (10 teams used the *Autonomous RPA,* 10 teams used the *Cooperative RPA* and five used the *Naive RPA*) from the University of Pittsburgh and Carnegie Mellon University communities. Participants were recruited as intact teams, consisting of friends or acquaintances. Each team member began at a different starting point, but all had the same rendezvous point. Teammates communicated with one another using electronic messaging to complete their tasks successfully. Results are reported for the 23 teams for which there is complete data.

5.3.2. Procedures

Each team participated in a 90-minute session that began with a 30-minute training session in which the *MokSAF* environment and team mission were explained. The team was told to find the best paths between the start and rendezvous points, to avoid certain areas or go by other areas, to meet the mission objectives for numbers and types of units in their platoon, and to avoid crossing paths with the other commanders. After the training session, the team participated in two 15-minute trials. Each trial used the same terrain, but different start and rendezvous points and different platoon requirements. At the conclusion, participants were asked to complete a brief questionnaire. We measured individual and team performance at the planning task, and analyzed communications among the team members.

5.3.3. Results

Data was examined from two critical points in the session – the time that individuals first shared their individual routes (first share) and at the end of the 15 minute session (final). Overall, we found that the two aided conditions, *Autonomous RPA* and *Cooperative RPA* achieved lower cost paths, earlier rendezvous, and lower fuel usage.

These results held true both for the team as a whole and for individual participants. It was expected that path lengths between the first time a route was shared and at the end of a trial would vary due to issues related to conflict resolution among the teammates. As shown in Figure 10, participants in the aided conditions managed to maintain the quality of their plans despite the modifications and replanning needed to coordinate with other team members. An ANOVA for total path length found a main effect for type of agent, F(2,20)=67.975, p < .001, (Naïve mean=552.5, SD=41.45; Autonomous mean=282.6, SD=52.06; Cooperative mean=260.22, SD=31.25) and post hoc tests using Tukey's HSD statistic showed both the Cooperative and Autonomous groups differed significantly (p<.001) from the Naïve control condition. A main effect was also found for total route times, F(2,20)=3.519, p =.049, (Naïve mean=2617, SD=358; Autonomous mean=2170, SD=275; Cooperative mean=2192, SD=215).

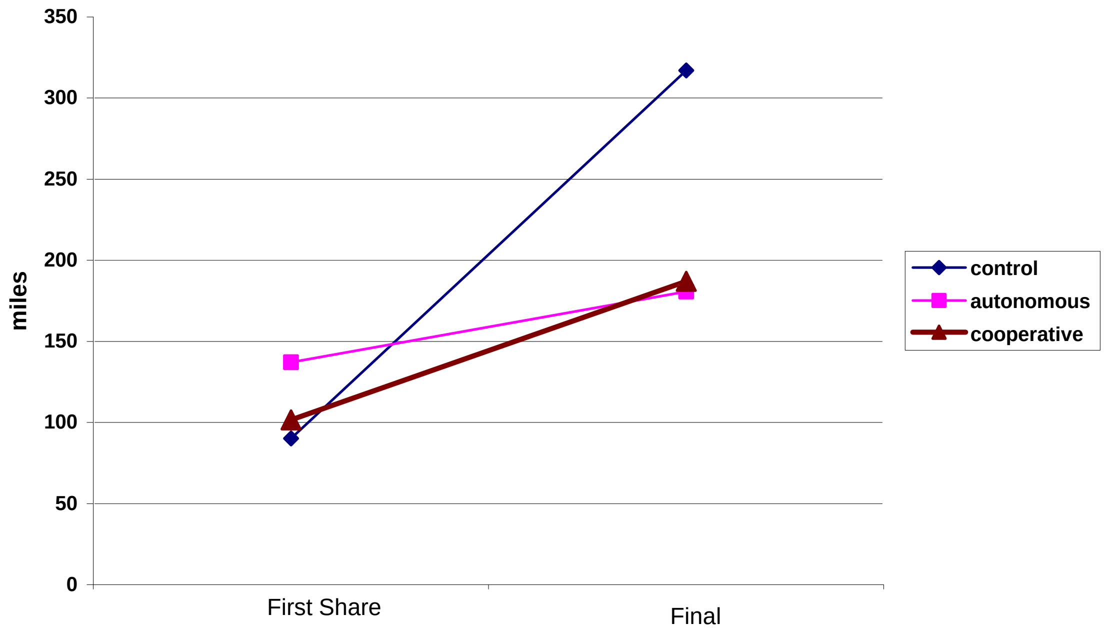

Figure 10. Path lengths were shorter for aided teams

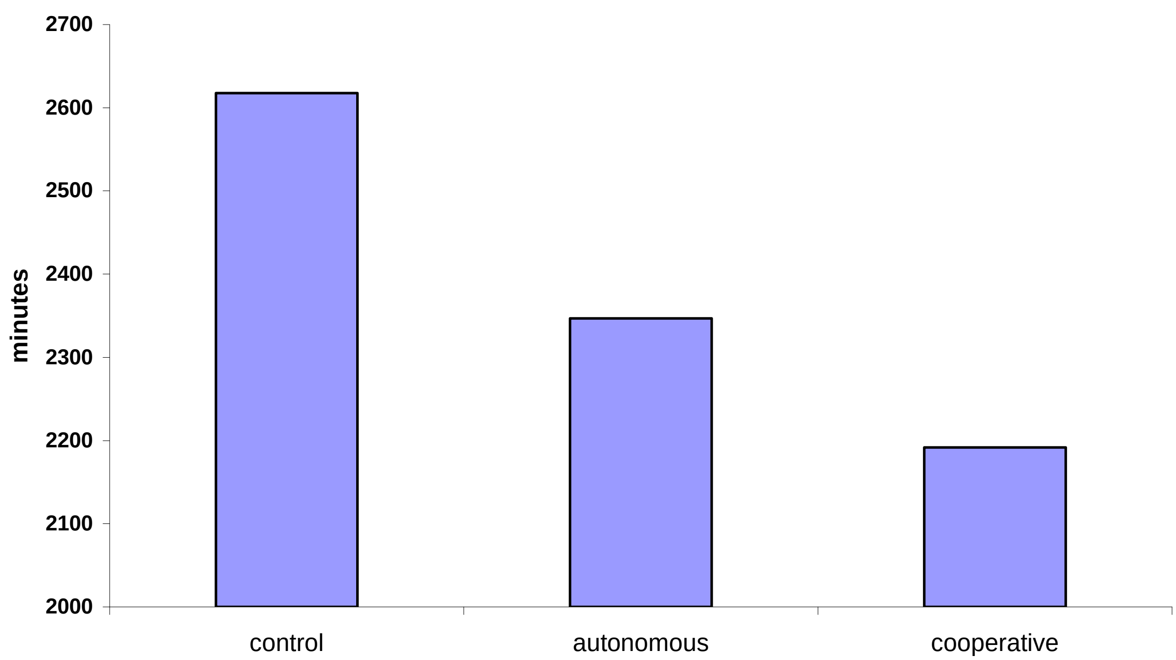

Figure 11. Travel times were reduced for aided commanders esp. Cooperative RPA users

As figure 11 shows the conditions retain a Cooperative, Autonomous, Naïve ordering although post hoc tests show the only significant difference (p=.04) to be between the Cooperative and Naïve agents. Although the RPA agents did not support teamwork directly, their assistance for the individual planning task allowed the commanders to find new routes as short as the ones they abandoned. Unaided commanders by contrast were forced to resort to longer paths in order to accommodate the requirements of coordinating with their team.

Teams participated in three sessions; the first session was training. The second session involved a more challenging task to correctly find an appropriate route from the starting point to the rendezvous point for all three commanders. They appeared to spend most of their time on this individual task and very little time on the team task of coordinating the selection of vehicles and meeting at the rendezvous point. On this more difficult Session 2 task, teams using the *Cooperative RPA* most closely approximated reference performance on the interdependent team task of selecting units. While the main effect for vehicle selection (sum of over and under represented unit types) only approaches significance, $F(2,20)=3.078$, $p=.068$; (Naïve mean=12.75, SD=6.99; Autonomous mean=9.1, SD=2.56; Cooperative mean=6.78, SD=3.87) as figure 12 shows the difference appears substantial.

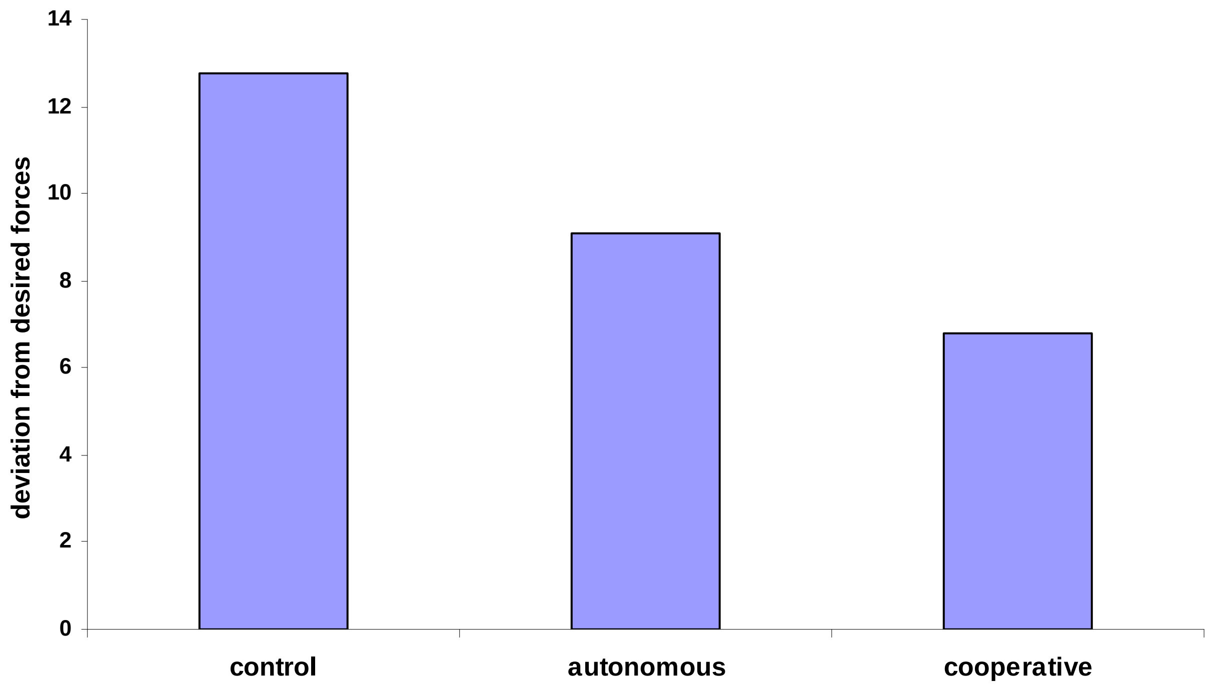

Figure 12. Agent help with route planning freed resources for vehicle selection

5.3.4. Challenges and Agent Effectiveness

Aided conditions were universally superior to the control in pathlength and fuel usage. Differences among the aided conditions arose due to differing levels of control over the precise path and more subtle differences in cognitive loading which allowed cooperative RPA subjects extra opportunity to consider team obligations.

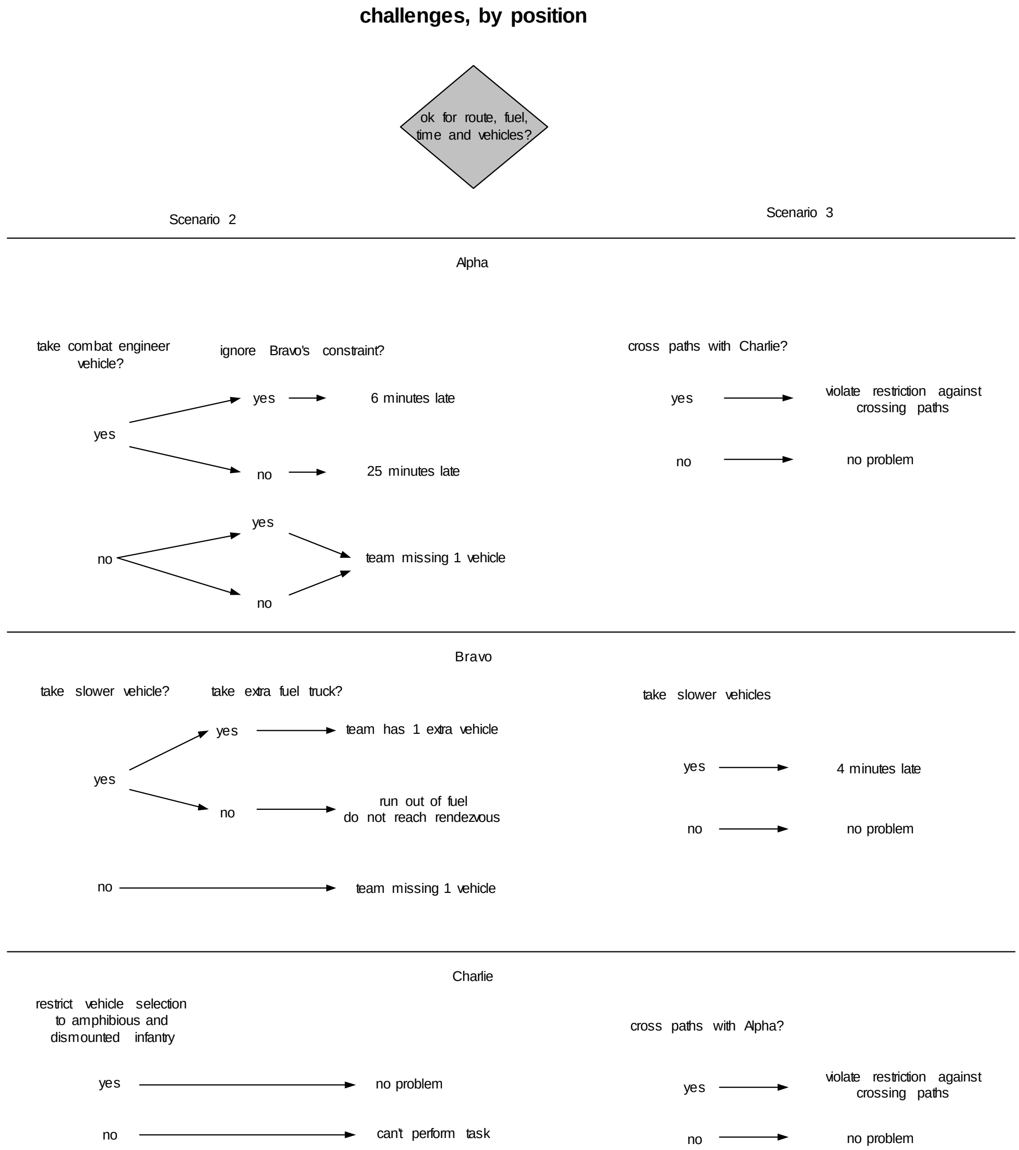

Figure 13. Challenges by position

In session 2 Alpha must decide to be altruistic by taking the combat engineer battalion which no one else could efficiently take. As a consequence Alpha must accept the responsibility for being 6 minutes late to the rendezvous. The obligation to take the engineers was not immediately apparent to subjects using the autonomous RPA . They did not discover the need to sacrifice their on-time arrival until after sharing paths and discussion the problem with other commanders.

For Bravo the complicating circumstance was the need to take a fuel truck in session 2. Subjects in the cooperative condition shared their path after discovering this necessary strategy but did not optimize. Subjects in the autonomous RPA condition were confronted with the need for a fuel truck early as their “optimal” routes kept failing. By the end of the session cooperative RPA subjects had achieved equally short routes.

In session 3 Charlie is confronted with the problem of contending with Alpha over his path. In the cooperative RPA condition participants in the Charlie condition resolved their problem without lengthening their route..

In the autonomous RPA the conflict proved to be a much more debilitating obstacle and was never satisfactorily resolved by many of the teams. Reported path lengths are for the teams in which Charlie had a legal route at session's end.

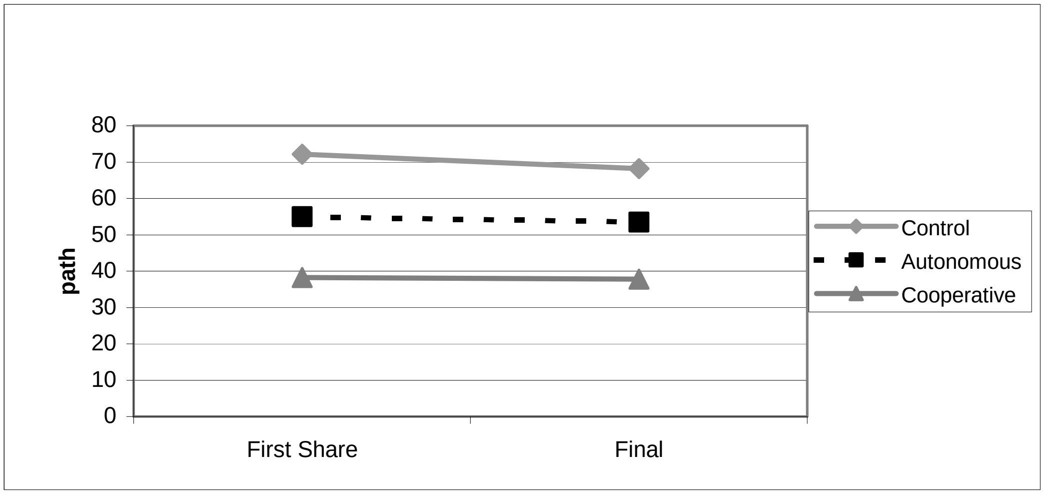

Figure 14. Path length for Charlie Session 3

5.3.5. Discussion

In its current form, the aided conditions, *Autonomous RPA* and *Cooperative RPA* have been shown to provide a better interface for both individual route planning and team-based re-planning. Despite this clear superiority over the unaided condition (*Naïve RPA*), participants in the *Autonomous RPA* group frequently expressed frustration with the indirection required to arrange constraints in the ways needed to steer the agent's behavior and often remarked that they wished they could "just draw the route by hand".

Comments on the *Naive RPA* focused more closely on the minutiae of interaction. In its current form, the user "draws" a route on the *interface agent* by specifying a sequence of points at the resolution of the terrain database. To do this, the user clicks to specify an initial or intermediate point in the path and then clicks again at a second point. A sequence of points is then drawn in a straight line between these locations. A route is built up incrementally by piecing together a long sequence of such segments. Although tools are provided for deleting unwanted points and moving control points, the process of manually constructing a long route is both tedious and error prone. While the *Cooperative RPA* automatically avoids local obstacles such as trees and closely follows curves in roads due to their less costly terrain weights, a user constructing a manual route is constantly fighting unseen obstacles which void her path or line segments which stray a point or two off a road into high penalty terrain. The anticipated advantages of heuristic planning and cooperation among human users were largely lost due to the necessity of focusing on local rather than global features of routes in the *Naïve RPA* condition. Rather than zooming in and out on the map to see the start and rendezvous points before beginning to draw, our subjects were forced to work from the first at the highest magnification in order to draw locally correct segments. The resulting problems of maintaining appropriate directions across scrolling segments of a map are not dissimilar to hiking with a compass. Although you can generally move in approximately the right direction you are unable to take advantage of features of the terrain you might exploit if a more global view were available.

While the Naïve RPA gratuitously forced the human to deal with physical constraints to which it already had direct access, the Autonomous RPA also diverted the user from the conceptual task of choosing and coordinating routes to that of representing intangible constraints. We believe this conceptual incongruity between the route planning and representation tasks left Autonomous RPA subjects unprimed for route planning related communications and coordination. Although the quality (path length/fuel usage) of routes between these two groups was very similar Cooperative RPA teams were better able to coordinate the composition of their forces and deal with deconflicting routes. These results emphasize the importance of designing human-agent interactions that promote direct interaction with the problem domain rather than focusing on information needs of the automation.

6. Conclusions

This paper draws together two complementary lines of research: the development of complex multiagent systems for aiding teams and factors that affect their acceptance and usefulness. As the amount of digital information available on the battlefield, to emergency response teams, and in our daily lives grows exponentially, intelligent assistance to filter, organize, and make sense of these seas of information is becoming a necessity. Whether this delegated information processing is done in an ad hoc fashion or reflects efficient design in allocation of function and exercise of control will depend on extending our scientific understanding of the problems of interaction among human-agent teams.

The ATOM model of teamwork suggests that teams could be aided along four general dimensions: 1) accessing information, 2) communicating, 3) monitoring and 4) planning[1]. Researchers in the information agent area have conventionally focused on information access, while those in the personal associate tradition (Banks and Lizza 1991) have concentrated on monitoring (and intervention). Our research suggests that there may be fruitful applications of agent technology to communications and planning as well.

The Tandem experiments illustrate the extent to which assistance in information exchange can have a major impact particularly in real-time high workload tasks. As we come to rely more and more on ad hoc interlocking teams made possible by the increasing interconnection and digitalization of the battlefield there will be fewer opportunities for team training and co-adaptation shown to be essential to high performance teams. For newly formed teams, such as our Tandem subjects, assistance in selecting and directing communications provided major benefits. The agent’s role of identifying salient changes in the tactical picture as humans uncovered them and relaying these updates to the

[1] The ATOM descriptors of “leadership” and “initiative” do not yet fit the team roles allowed agents. Current activities in this category center on recognizing failures in plans and missed planning opportunities (critiquing).

appropriate teammates is a new role for software agents. Rather than acting as personal agents assisting the information recipients by seeking this information (pull) or a personal agent assisting the information provider in finding information, the agent supports distribution (push) of their products. As networked communications become increasingly interconnected and ubiquitous as envisioned for the Air Force's Joint Battlespace Infosphere (Holzhauer et al. 2001) automation of these tasks of updating and distributing new information will be crucial to maintaining an electronic tactical picture. We believe that software agents must play a central role (Sycara and Lewis 2002) both in incorporating new information into an evolving picture and in providing a gateway for humans to augment, annotate, and enrich this tactical picture.

The closely related dimension of "communication" which characterizes the interpretability of communicated messages becomes a matter of good information presentation (agent-to-human) and design of effective interaction techniques (human-to-agent). Our Mobile Communications among Heterogeneous Agents (Mocha http://www.ri.cmu.edu/projects/project_425.html) project addresses the agent-to-human problem by tailoring information presentation to the available device and user's context. The planning dimension enters this mix through anticipatory information retrieval and intelligent caching featured in the Jocasta technology demonstration. Human-to-agent communication was investigated in our Route Planning experiments. Our findings highlight the importance of designing interactions that focus attention on the human's goals and problem domain rather than the agent's information needs. This need to tailor human-agent interactions to tasks and domain suggests that intelligent agents are likely to add to rather than detract from efforts needed in human-computer interface design. As software agents become more common in human teams we expect monitoring, correction, and intervention to become more acceptable but they are likely to be the last capabilities to be introduced into successful systems.

Already, much intelligence and general information can be gathered from the Internet whether for military operations or surgical ones. The most crucial real time data, however, remains trapped in stovepipe systems and standalone monitors and alarms. For software agents to truly contribute to human teams the common information available to other team members must be made available to them as well. As the movement toward integrating realtime and other organizational data gains ground the usefulness of agents will increase even more rapidly. An especially promising strategy my be to use agents such as our Interoperator (Giampapa et al. 2000) to assist in flexible interconnection of legacy systems.

Agent assistance will be particularly critical to military teams as their operations become more agile and situation specific. As unfamiliar forces are brought together for one-time missions the infosphere they establish between their networked information systems will become a primary mechanism for coordination. In this uncertain environment supporting teamwork becomes crucial. Our results suggest that software agents are well suited for this task. Because the domain independence of teamwork agents would allow them to be rapidly deployed across a broad range of tasks and settings teamwork appears to be a particularly high payoff area for further agent research.

7.References

Banks, S. & Lizza, C. (1991). Pilot's Associate: A cooperative, knowledge-based system application, IEEE Intelligent Systems and Their Applications, 6(3), pp. 18-29.

Beyerlein, M. M., Johnson, D. A., & Beyerlein, S. T. (Eds.). (2001). *Virtual teams. Advances in interdisciplinary studies of work teams*. Elsevier Science/JAI Press: Ukraine.

Chen, L. and Sycara, K. (1998). "WebMate: A personal agent for browsing and searching", In Proceedings of the 2nd International Conference on Autonomous Agents, Minneapolis, MN, May 10-13.

Cannon-Bowers, J. A., & Salas, E. (Eds.). (1998). Making decisions under stress: Implications for individual and team training. Washington, DC: APA.

Cannon-Bowers and Salas,.(1998). Individual and team decision making under stress: Theoretical underpinnings, in Cannon-Bowers. J. A. and Salas, E. (eds.) Making Decisions Under Stress: Implications for Individual and Team Training. American Psychological Association: Washington, D.C.

Cohen, P. R., and H. J. Levesque. (1990). Intention is choice with commitment, Artificial Intelligence 42(3), pp. 213-61.

Decker, K., Sycara, K. and Williamson, M. (1997). "Middle-Agents for the internet", Proceedings of the Fifteenth International Joint Conference on Artificial Intelligence (IJCAI-97), Nagoya, Japan, August 1997 pp. 578-584.

Decker, K., Pannu A., Sycara, K., and Williamson M. (1997). Designing behaviors for information agents, Proceedings of the First International Conference on Autonomous Agents (Agents-97), February, Los Angeles, CA., 1997.

Entin, E.E. and Serfaty, D. (1999). Adaptive team coordination. Human Factors, 41,2, pp.312-325.

Fiore, S. M., Salas, E., & Cannon-Bowers, J. A. (2001). Group dynamics and shared mental model development. In M. London (Ed.), How people evaluate others in organizations: Person perception and interpersonal judgment in Industrial/Organizational psychology (pp. 309 – 336). Mahwah, NJ: Lawrence Erlbaum.

Fiore, S. M., Salas, E., Cuevas, H. M. & Bowers, C. A. (in press). Distributed coordination space: Toward a theory of distributed team performance. Theoretical Issues in Ergonomic Science.

Foushee, H.C. (1984). Dyads and Triads at 35,000 feet: Factors affecting group process and aircrew performance. American Psychologist, 39, 885-893.

Giampapa, J., Paolucci M. and Sycara, K. (2000). Agent interoperation across multiagent boundaries", In Proceedings of the Fourth International Conference on Autonomous Agents, June 3-7, Barcelona, Spain, 2000. pp 179-186.

Grosz, B. and Kraus, S. (1996). Collaborative Plans for Complex Group Action. Artificial Intelligence journal, 86(2), pp 269-357.

Hess, K.P., Entin, E.E., Hess, S.M., Hutchins, S.G., Kemple, W.G., Kleinman, D.L., Hocevar, S.P., & Serfaty, D. (2000). Building adaptive organizations: A bridge from basic research to operational exercises. Proceedings of the 2000 Command and Control Research and Technology Symposium, June 26-28, 2000, Naval Postgraduate School, Monterey, CA.

Holzhauer, D., Combs, V., Linderman, M., Duncomb, J., Dyson, R., and Young, D.(2001). Building an experimental Joint Battlespace Infosphere (YJBI-CB). 6th International Command and Control Research and Technology Symposium, United States Naval Academy (USNA) in Annapolis, MD June 19 –21,. 2001.

Lee, J. and Moray, N. (1992). Trust, control strategies, and allocation of function in human-machine systems. Ergonomics 35(10), 1992., pp. 1243-1270.

Lee, J. and Moray, N. (1994). Trust, self-confidence, and operators' adaptation to automation. International Journal of Human-Computer Studies, 40, 153-184.

Lenox, T., Lewis, M. Roth, E., Shern, R., Roberts, L., Rafalski, T., and Jacobson, J. (1998). Support of teamwork in human-agent teams. Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, 1998, pp. 1341-1346.

Lenox, T., Roberts, L., and Lewis, M. (1997). Human-Agent interaction in a target identification task. In 1997 IEEE International Conference on Systems, Man, and Cybernetics, pages, Orlando, FL: IEEE, 1997, pp. 2702-2706.

Lewandowsky, S., Mundy, M., & Tan, G. P. A. (2000). The dynamics of trust: Comparing humans to automation. Journal of Experimental Psychology: Applied, 6, 104-123.

McNeese, M., Salas, E., & Endsley, M. (Eds.) (2001). New trends in collaborative activities: Understanding system dynamics in complex environments. Santa Monica, CA: Human Factors and Ergonomics Society.

Moray, N., Inagaki, T., & Itoh, M. (2000). Adaptive automation, trust, and self-confidence in fault management of time-critical tasks. Journal of Experimental Psychology: Applied, 6, 44-58.

Muir, B. (1994). Trust in automation: Part I. Theoretical issues in the study of trust and human Intervention in a process control simulation. Ergonomics, 37, 1905-1992.

Muir, B. and Moray N. (1996). Trust in automation: Part II. Experimental studies of trust and human Intervention in a process control simulation. Ergonomics, 39(3), 429-460.

Orasanu, J. & Salas, E. (1993). Team decision making in complex environments. . In G. Klein, J. Orasanu, R. Calderwood, & C. E. Zsambok (Eds.), Decision making in action: models and methods. Norwood, NJ: Ablex Publishing Company, pp. 327-345.

Paolucci, M., Onn Shehory and Sycara, K. (2001). "Interleaving planning and execution in a multiagent team planning environment". In the Journal of Electronic Transactions of Artificial Intelligence, May 2001.

Payne, T., Sycara, K. and Lewis, M. (2000). "Varying the user interaction within multi-agent systems" , In Proceedings of the Fourth International Conference on Autonomous Agents, June 3-7, Barcelona, Spain, 2000. pp 412-418

Salas, E., Bowers, C. A., & Edens, E. (Eds.) (2001). Improving teamwork in organizations: Applications of resource management training. Mahwah, NJ: Lawrence Erlbaum Associates, Inc.

Salas, E., Cannon-Bowers, J. A., & Johnston, J. H. (1997). How can you turn a team of experts into an expert team?: Emerging training strategies. In C. E. Zsambok and G. Klein (Eds.), Naturalistic decision making. Expertise: Research and applications (pp. 359-370). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc,

Salas, E.. Dickinson, T., Converse, S. & Tannenbaum, S. (1992). In R. W. Swezey & E. Salas (eds.), Teams: their training and performance. Ablex Publishing Corporation: Norwood, NJ, pp. 3 -29.

Shehory, O. and Sycara K. (2000). "The RETSINA Communicator", In Proceedings of the Fourth International Conference on Autonomous Agents, June 3-7, Barcelona, Spain, 2000. pp 199-200.

Smith-Jentsch, K. A., Zeisig, R. L., Acton, B. & McPherson, J.(1998a) Team dimensional training. In J. A. Cannon-Bowers & E. Salas (Eds.) Decision making under stress: Implications for individual and team training. Washington DC: American Psychological Association, pp. 271-312.

Smith-Jentsch, K. Johnston, J. H., and Payne, S. (1998b). Measuring team-related expertise in complex environments. In J. A. Cannon-Bowers and E. Salas (eds.), Decision Making Under Stress: Implications for Individual and Team Training. Washington, DC: American Psychological Association.

Sycara K., Decker, K., Pannu A., Williamson M., and Zeng D. (1996). Distributed intelligent agents, IEEE Expert: Intelligent Systems and their Applications, 2(6), pp. 36-46.

Sycara, K. and Zeng, D. (1996). Coordination of multiple intelligent software agents", International Journal of Intelligent and Cooperative Information Systems, 5(2-3) pp. 181-211.

Sycara, K., Klusch, M., Widoff, S. and Lu J.(1999). “Dynamic service matchmaking among agents in open information environments”, SIGMOD Record (ACM Special Interests Group on Management of Data), 28(1), pp. 47-53.

Sycara. K, Paolucci, M, Giampapa, J, and van Velsen, M. (2001). “The RETSINA MAS infrastructure” in the Journal of Autonomous Agent and Multiagent Systems.

Sycara, K. and Lewis, M. (2002). From data to actionable knowledge and decision. Proceedings of the Fifth International Conference on Information Fusion, Annapolis, MD, July 7-11, 2002.

Tambe, M. and W. Zhang, W. (1998). Towards flexible teamwork in persistent teams, In: Proceedings of the International Conference on Multi-Agent Systems (ICMAS), 1998.

Wong, H.C., and Sycara, K. (2000). “A Taxonomy of Middle-agents for the Internet”. In Proc. of the 4th International Conference on Multi-Agent Systems, 2000, pp. 465-466.