Task Characteristics and Intelligent Aiding

Terri Lenox, Michael Lewis, Susan Hahn
University of Pittsburgh
Pittsburgh, PA

Terry Payne & Katia Sycara
Carnegie Mellon University
Pittsburgh, PA

Abstract

In this paper, we describe the interactions between task characteristics and human agent interfaces in a team redezvous route-planning task. The agents include an interface agent and two different task agents that perform similar tasks. The MokSAF interface agent links an Artificial Intelligence (AI) route planning agent to a Geographic Information System (GIS). Through this agent, the user specifies a start and an end point, and describes the composition and characteristics of a military platoon. Two aided conditions and one non-aided condition were examined. In the first aided condition, a route-planning agent (known as the Autonomous RPA) determines a minimum cost path between the specified end points. The user is allowed to define additional "intangible" constraints that describe situational or social information that should be considered when determining the route. In the second aided condition, a different agent, the Cooperative RPA, uses the same knowledge of the terrain and cost functions available to the Autonomous RPA, but restricts its search to paths within regions drawn by the user. In the unaided condition, Naïve RPA, the user draws the route manually, then submits it to be tested against the terrain and cost functions for feasibility. Both aided conditions are superior to the control but differ in their relative effectiveness by scenario. In this paper we examine the varieties of challenges faced by commanders in two scenarios and relate them to the differential effectiveness of the agents.

1. Introduction

Demands on human decision-makers are rapidly increasing due to complex tasks and complicated technologies. As the task environment becomes more complex and uncertain and the time frame for making decisions is shortened, reliance on computer-assisted decision-making by both individuals and teams has become necessary To be successful, team members must understand how to gather, summarize, interpret, and use information to perform their task(s). In addition, team members must understand their role in the task and what information is required by their teammates. Finally, they should be aware of and act in accordance with the strengths and weaknesses of their teammates [1]. We have been examining different ways that software agents can be deployed in support of team performance [2] including:

- Support the individual team members in completion of their own tasks;
- Allocate to the agent its own subtask as if we were introducing another member into the team;
- Support the team as a whole

The present research investigates the effects of two distinctly different approaches to interacting with a route planning agent (RPA). The RPA assists individual commanders by finding a route between a starting point and the rendezvous. Support of the individual task of route finding could support team performance both by improving the routes of the individual commanders and by freeing cognitive resources allowing commanders to coordinate with one another more effectively.

One RPA agent is highly autonomous and relatively difficult to direct. The other RPA requires the commander to approximate the desired route and then finds the best route consistent with the approximation. In this paper we report findings relating the difficulty of commanders' tasks and type of RPA to individual and team performance.

1.1 Autonomous Intelligent Agents

In recent years, the approach used to solve complex problems has shifted from developing large, integrated legacy software systems, to that of developing small, autonomous, intelligent software components that can interact with humans, with other software components, and different data sources. These *agents* may provide specialized or periodic information from certain information sources, or may perform some task or service based on the information they are given. There are three general classes of agents: *interface agents, task agents,* and *information agents* [3]. *Interface agents* interact with the user, providing a mechanism whereby humans can specify tasks and inspect the results. They may acquire, model, and utilize user preferences to guide system coordination in support of the user's tasks [4]. *Task agents* help humans perform tasks by formulating problem solving plans, and carrying out these plans through querying and exchanging information with other software agents [5]. *Information agents* provide intelligent access to a heterogeneous collection of information sources [6].

The choice of task agent and the approach used by the interface agent to interact with the human can affect the behavior and utility of the agent team. Many interface agents determine user profiles to *personalize* the performance of an agent, or to improve the assistance provided by the agent [4]. In doing so, the agent may have to contact several different *task agents*, where each agent may perform very similar tasks. The choice of agent contacted will not only affect the overall performance of the team, but may require the agent to change the way in which it interacts with the human.

0-7803-6583-6/00/$10.00 © 2000 IEEE

1.2 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 are part of the commander's "infosphere." Information within the infosphere has the opportunity for data fusion, situation visualization, and "what-if" simulations. Software agents have access to all information in the infosphere and can plan, criticize, and predict the consequences of actions using the infosphere information at a greater accuracy and finer granularity than the human commanders can. Agent communities can be designed to use information cooperatively in the infosphere to satisfy specified goals.

Before the agents consider information that is outside the infosphere, this information should be captured in physical terms, and shared with the agents. This extra-infosphere data consists of intangible or multiple objectives involving morale, the political impact of actions (or inaction), intangible constraints, and the symbolic importance of different actions or objectives. Military commanders, like other 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). When participating in a planning task, commanders must translate these intangible constraints into physical ones to interact with planning agents.

The issue therefore becomes how should software agents interact with their human team members to incorporate these intangible constraints into the physical environment effectively.

2. The Planning Environment: *MokSAF*

A computer-based simulation called *MokSAF* has been developed 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 in 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.

The interface agent that is used within the *MokSAF* Environment is illustrated in Figure 1. 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 a 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.

2.1 Route-Planning Agents

Three different *route-planning agents* (*RPA*) have been developed which interact with the human team members in the planning task. The first agent, the *Autonomous RPA*, guides the human team members through the route-planning task and performs much of the task itself. This agent acts much like a

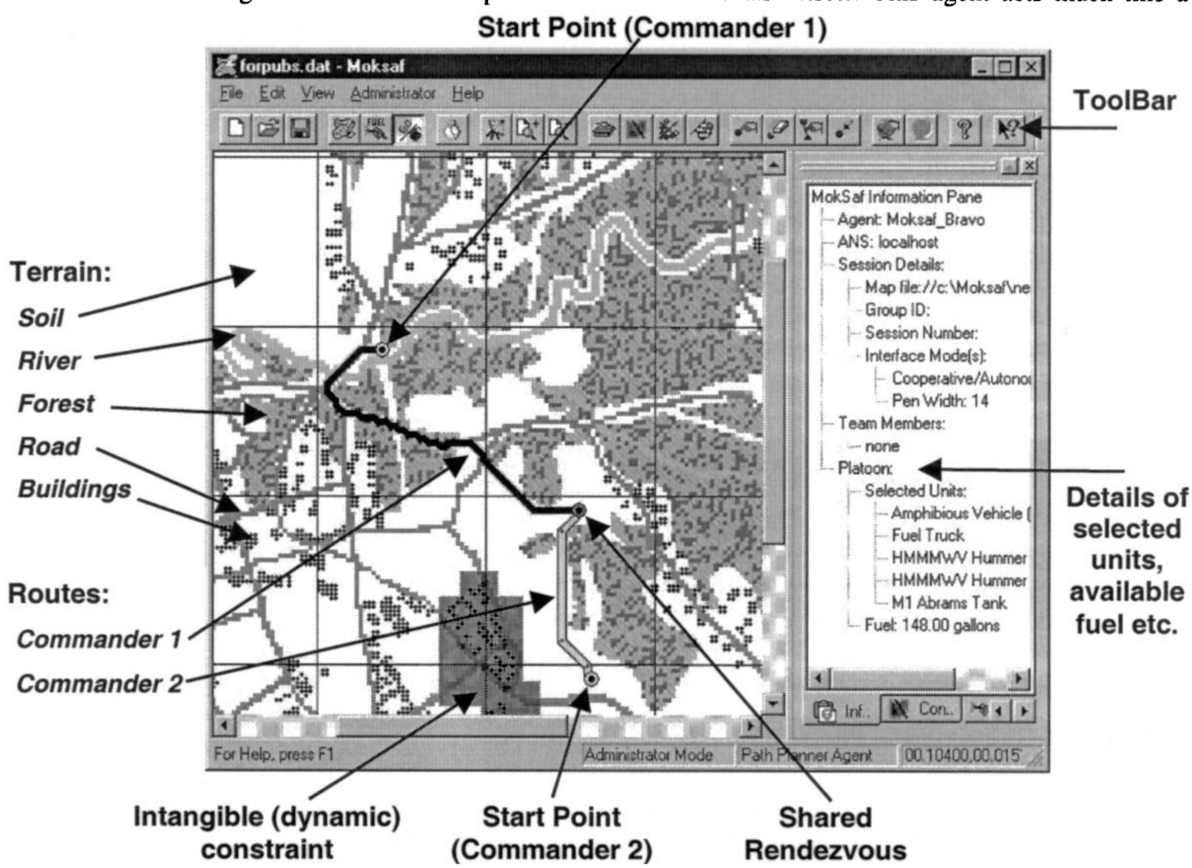

Figure 1: The MokSAF Interface Agent

"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 specified by the commanders.

The second agent, the *Cooperative RPA*, analyzes the routes drawn by the human team members, selects the optimal points within that 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 physical constraints.

The commander draws the desired route and requests that the *Cooperative RPA* reviews the route for physical violations or to indicate ways in which the path could be improved. The commander can iteratively improve the plans until a satisfactory solution is reached.

In the third condition, the *Naïve RPA*, provides minimal assistance to the human commanders in their task of drawing and refining routes.

3. Experimental Methodology

In the *MokSAF* pilot experiments, a deliberative, iterative and flexible planning task is examined. 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 one of the RPAs, each commander plans a route from a starting point to the rendezvous point for the specified platoon.

Once a commander is satisfied with the individual plan, he/she can share it with the other commanders and resolve any conflicts. Conflicts can arise due to several issues including shared routes and/or resources and the inability of a commander to reach the rendezvous point at the specified time. The commanders must coordinate regarding the number and types of vehicles they are planning to take to the rendezvous point. The mission supplied to the commanders provides them with a final total of vehicles required at the rendezvous point. In addition, the commanders are told that they should not plan a route that takes them on the same path as any other commander and that they should coordinate their routes to avoid shared paths.

3.1 Participants

Twenty five teams consisting of three-persons were recruited (10 teams who used the *Autonomous RPA*, 10 team who used the *Cooperative RPA* and five who 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 had a different starting point, but all had the same rendezvous point. Teammates needed to communicate with one another to complete their tasks successfully.

3.3 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.

4. Results

We examined the three experimental groups at two critical points in the session – 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 (see Figure 2)
- Earlier Rendezvous
- Lower fuel usage

than in the Naïve condition (unaided). These results held true for the team as a whole and for individual participants.

The two experimental scenarios (sessions 2 and 3) differed in difficulty and challenges presented to the commanders. In the more difficult session 2, Alpha should choose to take the fuel guzzling *combat engineer* and several slower vehicles which will cause him to be marginally late to the rendezvous in order to achieve best team wide performance. Bravo must take a fuel truck and some of the slower units, as well, for the team to reach rendezvous with a full complement of forces. Charlie must restrict unit selection to amphibious units and troops because land routes will not allow her to move other units to the rendezvous as efficiently as the other commanders.

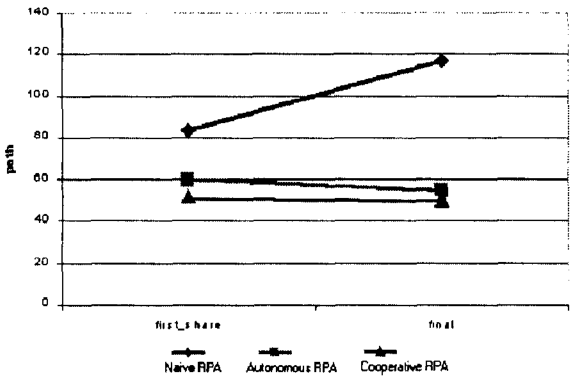

Figure 2: Path length was shorter in aided teams.

The commanders appeared to spend most of their time on the individual path planning 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 (Figure 3).

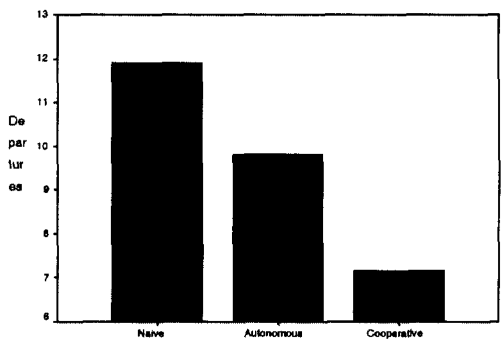

Figure 3: Departures from optimal vehicle selection were higher in the unaided teams *(Naïve RPA)*.

Teams using the *Autonomous RPA* made slightly less appropriate decisions. Finally, for this second session, teams using the *Naïve RPA* (unaided) performed poorly sometimes failing to rendezvous with teammates.

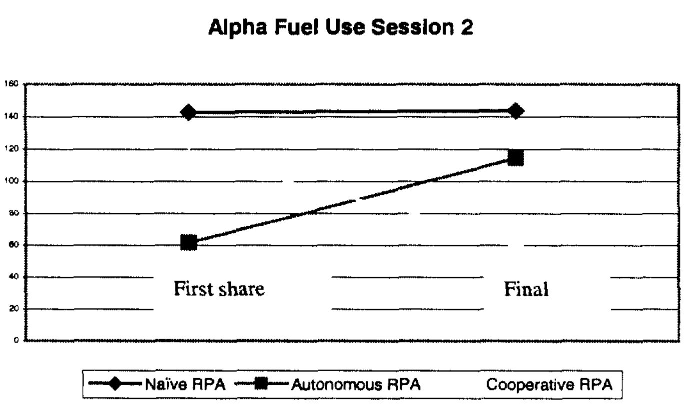

Figure 4 Alpha Session 2

Path lengths were roughly equal for autonomous and cooperative RPAs (Figure 2). For Alpha, the commander with the most complex choices in unit selection, fuel usage was affected by type of agent with team conscious commanders using the cooperative RPA selecting the higher fuel consuming forces needed to achieve proper force composition at rendezvous in their initial plans, then improving this route through the session.

The simpler third scenario pits Alpha against Charlie in a path crossing conflict. To successfully reach rendezvous without violating their briefs, one of these commanders must choose a route other than the shortest path. Unlike force composition which was often deficient, path crossing was largely avoided. As shown in Figure 5, cooperative RPA users made these adaptations with slightly greater efficiency. The difference in fuel efficiency for Charlie's position (Figure 6) reflects the "best choice" of path change for this position followed by cooperative RPA users.

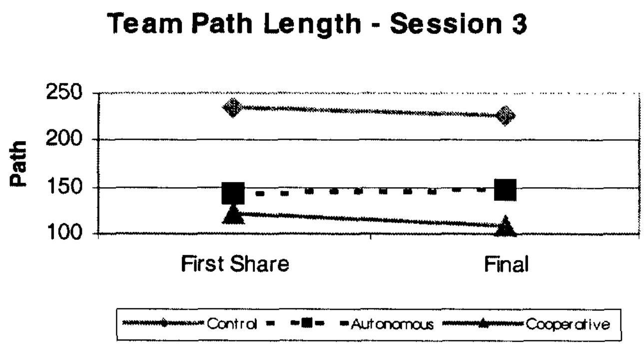

Figure 5 Session 3 Path Lengths

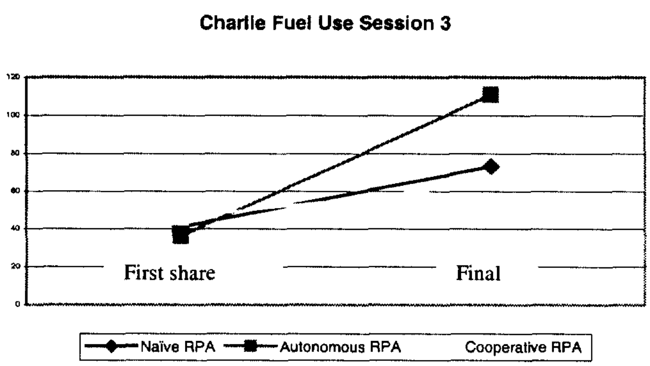

Figure 6 Charlie Session 3

Task difficulty was measured by:

- map grids crossed by reference path
- number of reversals in reference path

Participants receiving assistance from agents were found to:

- Communicate *less* during the initial individual planning phase
- Communicate *more* during the later coordination phase
- Develop **Team Plans** without sacrificing efficiencies of their individual plans

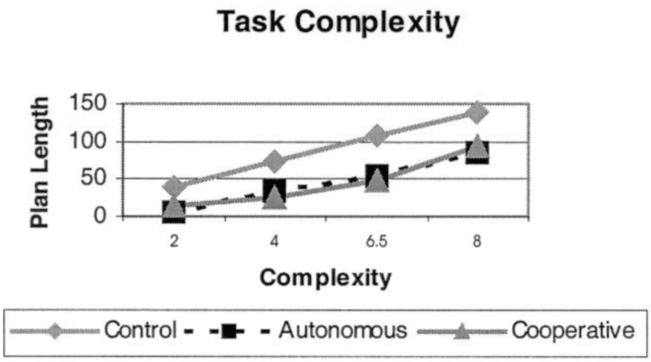

Figure 7 Complexity and path length

Figure 7 plots path lengths against complexity (reversals & grid crossings) for individual tasks (Alpha & Bravo session 2 and Bravo & Charlie session 3 have equivalent path complexities).

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. A route is built up incrementally by piecing together a long sequence of such segments making the process of manually constructing a long route is both tedious and error prone.

While autonomous and cooperative aiding strategies were equally successful in improving individual task performance as measured by quality of routes (path length, time, and fuel usage), the cooperative RPA appears superior in supporting team tasks. This is seen in scenario 2 where Alpha commanders using the cooperative RPA sacrificed their own on-time arrivals in order to bring units their fellow commanders could not. In scenario 3 we see a similar departure from local optimality in path planning to satisfy the team goal of avoiding path crossings.

Whether this superiority of cooperative aiding is a function of these particular agents and scenarios or a more general principle remains to be demonstrated. There were no significant differences in communications among the three conditions other than a slight edge in leadership (coordinating) messages in the cooperative RPA condition. Despite this rough equality in team work relevant information cooperative RPA users were able to use it more effectively. The closeness of the aided conditions in all other respects suggests that the improvement may not be due to cognitive resource limitations for autonomous RPA users but rather to the coupling of route planning, unit selection, and intersection avoidance in the cooperative condition. While the autonomous RPA makes these tasks independent, the participation required by the cooperative RPA provides opportunities for interactions between force selection and route planning to be observed and considered. We believe this loss of situational awareness may be a significant hazard in integrating intelligent agents into human teams and believe that strategies for involving human operators in automated decision making may be an important way to combat it.

Acknowledgements

This research was supported by an Office of Naval Research grant N-00014-96-1-1222. Special thanks go to Constantine Domashnev, Glenn Lautenbacher and Martin van Velsen for their assistance in the development of this software.

References

1. 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.

2. Lenox, T., Lewis, M. Roth, E., Shern, R., Roberts, L., Rafalski, T., and Jacobson, J. (1998). "Support of Teamwork in Human-Agent Teams." IEEE Conference on Systems, Man, and Cybernetics, 1341-1346.

3. Sycara, K., Decker, K., Pannu, A. S., Williamson, M. and Zeng, D. (1996). "Distributed Intelligent Agents." IEEE Expert 11, 6 (December 1996), 36-40.

4. Payne, T. R, Edwards, P., and Green, C. L. (1997). " Experience with Rule Induction and k-Nearest Neighbour Methods for Interface Agents that Learn, " IEEE Transactions on Knowledge and Data Engineering, 1997, Vol. 9(2), 329-335.

5. Paolucci, M., Shehory, O., Sycara, K., Kalp, D. and Pannu, A. A. (1999). "Planning Component for RETSINA Agents, " Lecture Notes in Artificial Intelligence, Intelligent Agents VI. M. Wooldridge and Y. Lesperance (Eds.).

6. Decker, K., Sycara, K.,Pannu, A., and Williamson, M. (1997). "Designing Behaviors for Information Agent," Proceedings of the First International Conference on Autonomous Agents (AGENTS-97), Feb., 1997.