

# AGENT-BASED AIDING FOR INDIVIDUAL AND TEAM PLANNING TASKS

Terri L. Lenox\*, Terry Payne\*\*, Susan Hahn\*, Michael Lewis\* & Katia Sycara \*\*

\*University of Pittsburgh  
School of Information Sciences  
135 N. Bellefield Ave.  
Pittsburgh, PA 15260  
[tl@lis.pitt.edu](mailto:tl@lis.pitt.edu)

\*\*Carnegie Mellon University  
Robotics Institution  
5000 Forbes Avenue  
Pittsburgh PA 15213 USA  
[terry.payne@cs.cmu.edu](mailto:terry.payne@cs.cmu.edu)

*Intelligent aiding strategies were evaluated for a team planning task. 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 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. In the second aided condition the same knowledge of the terrain is used to plot the best route within bounds specified by the user. In the control condition the user draws a route without help. The reported study found across the board advantages for agent-based aiding when routes were merged in a team-planning task.*

## INTRODUCTION

Our research focuses on techniques, which enable us to communicate with software agents. While much of the early focus on decision aids has been on supporting the individual [1], we examine the middle ground of individually controlled software agents used in team tasks.

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. 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. This choice 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.

Human decision-makers often 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 forces and other physical information stored electronically or available through sensors makes up the “infosphere.” Software agents can plan, criticize, and predict the consequences of actions using infosphere information at a greater accuracy and finer granularity than a human commander.

For agents to consider information from outside the infosphere, it must be expressed in a compatible form. When

cooperating in a planning task, a commander’s greatest effort may go into translating these intangible constraints into physical ones to take advantage of the agent’s expertise in planning within the infosphere.

## 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.

### 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 “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 by drawing prohibited areas on the map.

The second agent, the *Cooperative RPA*, analyzes approximate routes drawn by the human team members, and selects the optimal path within those bounds. 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 both human and machine agent are matched to their 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. While the agent can construct a path which is optimal with respect to the physical constraints of the infosphere.

In the control condition, the *Naïve RPA*, provides minimal assistance to the human commanders by reporting violations of physical constraints in proposed routes.

### 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

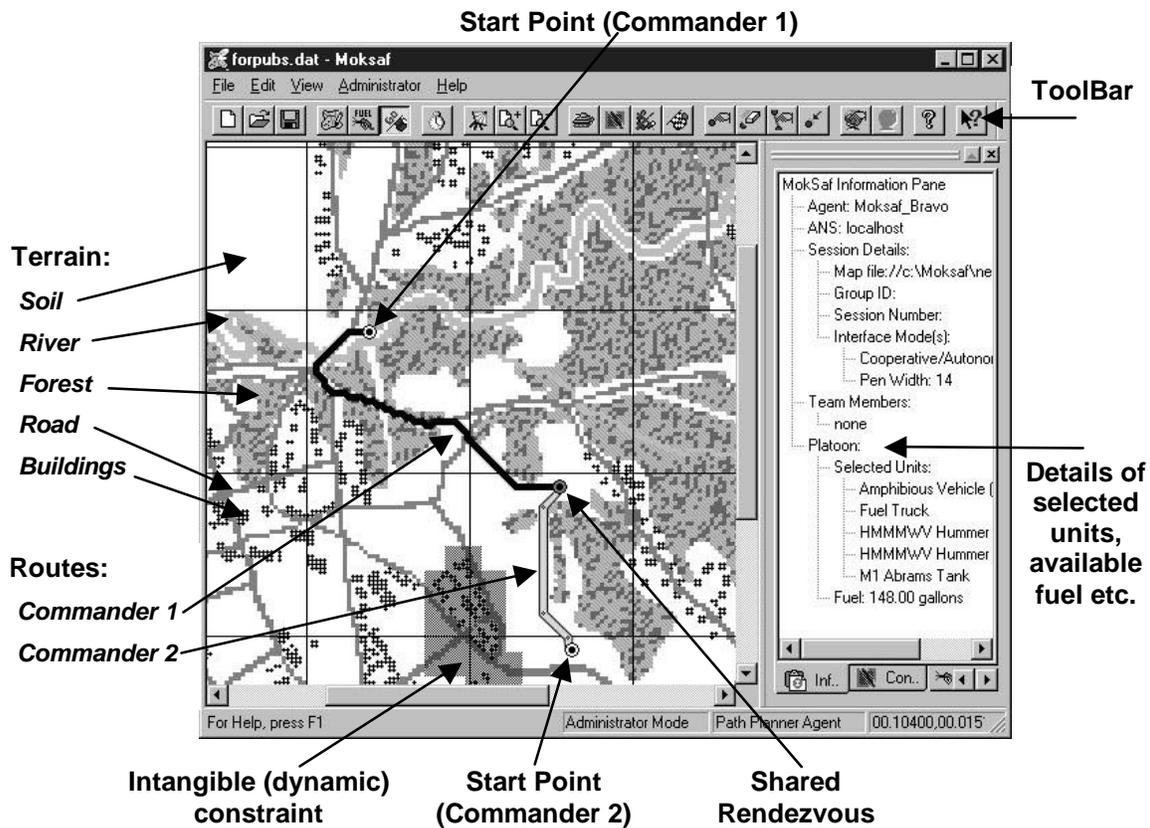


Figure 1: The MokSAF Interface Agent

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.

*MokSAF* 1.0 was used for this study. It consists of the standard terrain map and markings, a toolbar as seen in Figure 1, a communication window where commanders can send and receive messages and share plans, and a constraint tree. The three different agent interfaces described above were evaluated.

## Participants

Twenty five teams consisting of three-persons were recruited (10 teams who used the *Autonomous RPA*, 10 teams 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.

## 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 optimal path 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 are measuring individual and team performance with respect to the planning task, and using a cognitive task analysis technique will analyze the communications among the team members to determine if and how each type of agent supports the team as a whole. One question we hope to answer is which interface type best supports the overall team performance in this type of task. There are two expected trade-offs between the *Autonomous RPA* (which acts as an oracle) and the *Naive RPA* (which acts as a critique):

- 1) The complexity of intangible constraints and multiplicity of goals;
- 2) The time and/or quality of the agent-generated solutions (*Autonomous RPA*) versus the agent-critiqued solutions (*Naive RPA*).

## 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
  - Earlier Rendezvous
  - Lower fuel usage
- than in the *Naive* condition (unaided). These results held true for the team as a whole and for individual participants.

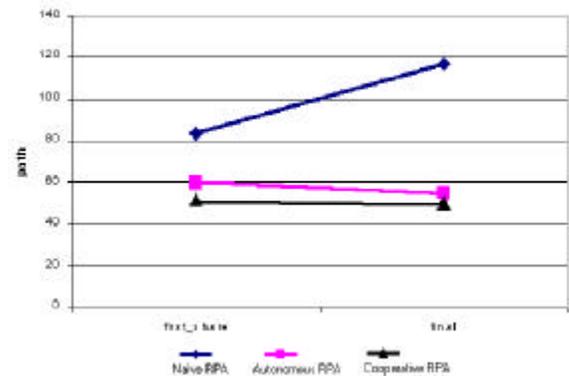


Figure 2: Path length was shorter in aided teams.

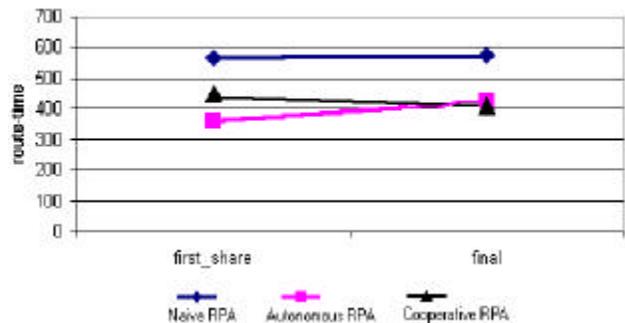
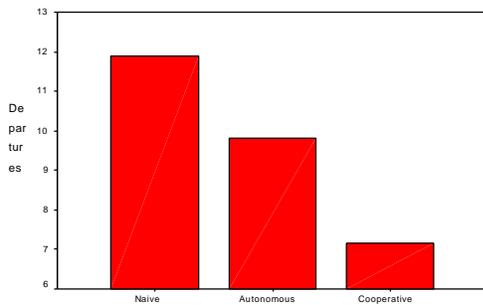


Figure 3: Route times were shorter for aided teams.

Teams participated in three sessions; the first session (1) was training. On the more difficult Session 2 task, teams using the *Cooperative RPA* most closely approximated reference performance (Figure 4).

Teams using the *Autonomous RPA* made slightly less appropriate decisions. Finally, for this second session, teams using the *Naive RPA* (unaided) performed poorly sometimes failing to rendezvous with teammates. For the less difficult Session 3 task, the team performance retained ordering although differences between aided conditions were not significant.



**Figure 4: Departures from optimal vehicle selection were higher in the unaided teams (session 2).**

The effects of aiding teams through the use of a software agent interface were consistent over the range of task difficulty as measured by: map grids crossed by reference path or number of reversals in reference path. Participants receiving assistance from agents:

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

## DISCUSSION

The *Autonomous RPA* and *Cooperative RPA* were shown to provide better interfaces for both individual route planning and team-based re-planning. Difficulties with the control condition (*Naive RPA*) appeared to be due to the minutiae of specification required to interact with infosphere data directly.. The user “drew” a route by specifying a sequence of points at the resolution of the terrain database. To do this, the user clicked to specify an initial or intermediate point in the path and then clicked again at a second point. A sequence of points was then drawn in a straight line between these locations. A route was 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 interaction with the *Autonomous* or *Cooperative RPA* automatically avoided local obstacles such as trees and closely followed curves in roads due to their less costly terrain weights, a user constructing a manual route must constantly fight unseen obstacles which voided the path or line segments which strayed off a road into high penalty terrain.. Rather than zooming in and out on the map to see the start and rendezvous points before beginning to draw, control 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.

Although the differences between the aided conditions and the control were far greater than those between each other, some differences were noted. In the difficult second session, no team in the control condition had a valid plan for rendezvousing by the time plans were first shared and only two out of five constructed valid plans by the end of session.. Finding valid routes in session 2 was difficult for *Cooperative RPA* subjects as well with several groups lacking complete plans at first sharing. Despite these early difficulties teams using *Cooperative RPA*'s. all constructed successful plans with quality comparable to *Autonomous RPA* teams by session's end. On the non-aided task of force selection *Cooperative RPA* users performed substantially better than *Autonomous RPA* teams in session two. We believe this superiority in force selection to be attributable to the greater user control over routes offered by the Cooperative RPA. By simplifying the task of altering their routes to accommodate one another's constraints they were encouraged and rewarded for communicating and able to devote more resources to the cooperative force selection task. Although differences between aiding conditions were not significant in session 3, participants in each position found shorter paths using the *Cooperative RPA*.

These results suggest that no single strategy for allocating control in intelligent aiding will be superior under all conditions. The autonomous agent was best in the difficult second session under time pressures to generate a first plan. If teams had needed to proceed on this basis, the autonomous agent would be the only acceptable choice. Where subsequent communication and coordination were allowed, however, the greater controllability of the cooperative agent proved to be essential both for more closely coordinating routes and easing the task of coordinating vehicle selection.

All three conditions organized and presented the same infosphere data to the participants. The two agents providing planning assistance offered users different interfaces to the same route-planning algorithm. We believe that this approach of providing a range of levels of autonomy and delegation for interacting with intelligent agents illustrates a general solution to the problem of shared control

## 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. Roth, E. M., Malin, J. T. and Schreckenghost, D. L. (1997). “Paradigms for Intelligent Interface Design.” In M. Helander, T. K. Landauer, and P. Prabhu (Eds) Handbook of Human-Computer Interaction, Second Edition, 1177 -1201.