

Agent-Based Support for Human/Agent Teams

Terry R. Payne[†], Terri L. Lenox[‡], Susan Hahn[‡], Katia Sycara[†], Michael Lewis[‡]

Robotics Institute[†]
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15232 USA
+1 (412) 268 8780
[terryp|katia]@cs.cmu.edu

School of Information Sciences[‡]
University of Pittsburgh
135 N. Bellefield Avenue
Pittsburgh, PA 15260
+1 (412) 624 1178
[tll|hahns|ml]@lis.pitt.edu

ABSTRACT

In this paper, we present an interface agent, *MokSAF*, which facilitates time-critical team-planning tasks for teams of both humans and heterogeneous software agents. This agent assists in the formation of teams of humans (via other *MokSAF* agents) and *task* agents that can autonomously perform team subtasks. It provides a suitable interaction mechanism to instruct the various task agents in the team; and, by monitoring the human's progress, reallocate or modify the sub-tasks if the human fails to achieve that sub-task. A military domain has been used to investigate this interface agent. The task consists of three military (human) commanders that each assemble a platoon, and plan routes so that all three platoons arrive at a given rendezvous by a specified time. An experimental study has been conducted to evaluate *MokSAF* and the assistance provided by one of three different *task* agents, and the results summarized.

Keywords

Interface Agents, Functional Substitutability, Multi-Agent Systems, Team Coordination

INTRODUCTION

Environmental uncertainties, stress, and time criticality of decision making characterize emergency response tasks (both military and civilian). The decision-making process may be distributed across different team members with different expertise, who are distributed in space and time, and who act with incomplete information within an uncertain environment. Hence, high quality computer assistance is critical. Recently, the technology of software agents has emerged as a suitable metaphor for interacting with computer processes that assist human decision-making. Such software agents can reduce the amount of interaction between humans and the computer system and allow the humans to concentrate on other activities, such as assessing the situation, making decisions, or reacting to changes in the system [7]. In addition, these agents not only retrieve

information on request; but actively and intelligently adapt, anticipate, and actively seek ways to support users [1,6].

When interacting with a computer, some metaphor should be adopted to guide the actions and expectations of the user [4]. The metaphor presented in this paper treats the computer as an intermediary that responds to user requests. Instead of simply entering commands or selecting objects from a GUI environment, the user and the agent initiate communication, execute tasks, and monitor their respective performances. This agent metaphor has been referred to as indirect management [4,5] and has been employed by the *MokSAF* environment.

THE MOKSAF ENVIRONMENT

The *MokSAF* environment allows two or more humans (acting as military commanders) to collaborate via two or more interface agents when planning missions. A mission plan consists of one or more platoons of heterogeneous units, an agreed rendezvous time and location, and a set of routes for each platoon. The platoons start from different points, and each route ends at a common rendezvous by a given time. Each commander must select appropriate units to constitute his/her platoon so that:

- (1) The platoon should reach the shared rendezvous point without running out of fuel.
- (2) The platoon should visit certain mid-points en-route.
- (3) The route should not violate any physical constraints.
- (4) The route should not violate intangible constraints.
- (5) The combined platoons should contain a minimum subset of specified units at the rendezvous.

Commanders are responsible for the composition of their platoon and for determining the route taken by that platoon. They can obtain assistance in determining a route from a task agent known as a *Route Planning Agent* (RPA). In addition, each commander can share his/her individual route with other commanders via the *MokSAF* agent.

If computerized aiding is to be effective within this domain, *intangible constraints* (which may be transient and unstructured, such as “when on an exercise avoid routes that go near schools during term time, unless the platoon consists of light vehicles”) should be encoded in a form that

can then be utilized by the agents. Some form of feedback should also be provided so that the commander can verify that the encoding is correct. To resolve this problem, intangible constraints are represented by shaded rectangles drawn on the map (displayed by the MokSAF agent). These regions represent areas that the platoon should avoid. This visual representation is easy for the user to both perform and verify. Once drawn, these constraints can be shared with an RPA, which can then utilize this knowledge when providing assistance with future routes.

Route Planning Agents (RPAs)

Three different methods for determining individual routes have been investigated using three different RPAs: *Naive RPA*, *Cooperative RPA*, & *Autonomous RPA*. These agents assist the commanders by assisting with one aspect of each of the commander's task, and thus allowing the human to focus on achieving the correct composition of units at the rendezvous. The *Naive RPA* analyses a commander's route, and provides basic information pertaining to fuel consumption and estimated travel time. The *Cooperative RPA* analyses a commander's route and refines the route to reduce fuel consumption and travel time. The *Autonomous RPA* identifies the optimal route between a specified start and rendezvous location. In each case, the MokSAF agent and RPA exchange knowledge about intangible constraints and the composition of units in the commander's platoon.

The three Route Planning Agents consult the same terrain map that is displayed by the MokSAF agent. Each pixel on the map corresponds to a traversal coefficient, depending on the terrain type and the speed characteristics of a given platoon. These speed characteristics are determined by finding the slowest speed of each of the units within that platoon (i.e. the platoon may travel no faster than its slowest unit). As certain units may only traverse certain terrain types, this may constrain the route of the final path.

The Cooperative and Autonomous RPAs utilize an off-road route-planning algorithm based on Dijkstra's shortest path algorithm [2] to generate or refine routes. This is used to determine the minimum cost route between two points (a start and rendezvous location). The routes sent to the Naive and Cooperative RPAs consist of an arbitrary number of points on the map, connected by straight-line segments. The Naive RPA checks to see whether this route violates any physical constraints (such as land based units crossing water) or encoded intrinsic constraints. If any such constraints have been violated, or if the platoon has insufficient fuel to complete the journey, the commander is notified, and violations/fuel status information is displayed on the MokSAF map.

Functional Substitutability

In certain scenarios, the commander may encounter difficulties in using a given RPA, such as repeatedly violating geographic constraints with the Naive RPA, or having to define several different intangible constraints when using the Autonomous RPA. The MokSAF agent

monitors the commander's progress, and if necessary advises the commander to use a different RPA to successfully determine the best route. Refinements to the platoon composition are also suggested if several routes are generated that require more fuel than is available.

The MokSAF environment has been used to investigate human team dynamics with the military domain described above. Groups of three subjects were recruited to perform the team task with each of the different RPAs [3]. The results thus far suggest that humans focus on the problem of determining the route (despite the varying levels of assistance provided by the RPAs) and neglect to improve the composition of the platoons.

CONCLUSIONS

The MokSAF agent is an interface agent that assists human commanders in coordinating and achieving team goals for mixed human/agent teams. Three different Route Planning Agents exist that can assist the commanders in a variety of ways. Should the commander fail to achieve his/her individual goal, the MokSAF agent will automate parts of the goal to assist the commander. To date, several experiments with human teams and different Route Planning Agents have been investigated [3].

ACKNOWLEDGMENTS

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

REFERENCES

1. Bradshaw, J.M. Introduction. In J.M. Bradshaw, (Ed.) *Software Agents*, AAAI Press, Menlo Park, CA (1997), 3-48.
2. Dijkstra, E.W. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1959), 269-271.
3. Lenox, T., Payne, T., Hahn, S., Lewis, M. and Sycara, K. MokSAF: How should we support teamwork in human-agent teams? Tech Report CMU-RI-TR-99-32, Robotics Institute, CMU, September, 1999
4. Lewis, M. Designing for human-agent interaction. *AI Magazine*, (Summer 1998), 67-78.
5. Maes, P. Agents that reduce work and information overload. *Commun. ACM* 37,7(1994), 31-40, 146.
6. Sycara K., Decker, K., Pannu A., Williamson M., and Zeng D. Distributed Intelligent Agents. *IEEE Expert: Intelligent Systems and their Applications*, Vol. 11, No. 6 (December 1996).
7. Zachary, W., Le Mentec, J-C. and Ryder, J. Interface agents in complex systems. In C. A. Ntuen and E. H. Parks (eds.) *Human Interaction with Complex Systems: Conceptual Principles and Design Practice*. Kluwer Academic Publishers (1996).

