Mixed-Initiative Agent Team Formation, Tasking and Management

Mark H. Burstein and David E. Diller

BBN Technologies
10 Moulton Street
Cambridge, MA 02138
burstein@bbn.com, ddiller@bbn.com

Abstract

Our interest is in the development of mixed-initiative interaction techniques to support heterogeneous human-agent organizations. If software agents are to become more commonplace, they will need to fit into human organizations at many levels, and users will need to direct and control teams of standalone agents and other humans supported by agent assistants. We have developed a demonstration of this kind of situation called MIATA (Mixed-Initiative Agent Team Administration) in collaboration with a group of other DARPA researchers. The demonstration scenario, based on the historical record of the U.S. effort to provide relief to Honduras after Hurricane Mitch, shows mixed human/agent teams forming to respond to the disaster. It illustrates several key ways in which intent reasoning must play a role in the functioning of agents that operate in such an organizational environment. In particular, we discuss the process of interactive team formation and a mechanism for dynamic information sharing among agents, each tasked by users to collaborate on teams.

Introduction

We are now facing a time where the informational and computational requirements of organizations such as the US Military, NASA, the FAA, and other large corporations require a synergistic coupling between humans and software systems. As their benefits increase, software agents will be increasingly integrated into existing organizational systems in order to perform a wide variety of tasks well suited to agent systems. We need to develop compatible methods of addressing both interactions between humans and software agent systems and the resulting inter-agent communications motivated by the intentions of the organizations of which they are a part. Our interest is in the development of mixed-initiative collaboration tools to support humans in mixed organizations of humans and software agent systems (e.g., Burstein and McDermott 1996; Burstein, Mulvehill, and Deutsch 1998, Burstein and Diller, forthcoming). These tools must help in organizing teams of agents, the tasking of those agents, and the handling of the unexpected situations and failures that are bound to arise.

This paper addresses several issues raised by a recent effort we led to develop a large-scale simulation of an organization supported by agents tasked with providing relief after Hurricane Mitch in 1998. Our approach is to simultaneously build mixed-initiative agent systems and an environment in which we can experimentally explore the requirements for human-agent and agent-agent interaction that facilitate collaborative decision-making within realistically complex domains.

As many people (e.g., Norman 1997) have pointed out, human users of agent systems are unlikely to develop trust in the capabilities of the agent until the agent can be employed much like a human subordinate, in a way that can be carefully monitored and managed. Agents must be able to respond effectively to task directives, collaborate with other agents within the organization, and handle a variety of failure conditions appropriately in order to be completely accepted. Until then, their role will be quite limited. This critically means agents must be able to make use of information about user and organizational intent in their attempts to do cooperative behaviors. We present several examples of ways that this could happen.

MIATA Simulation

For the past two years, we have been working with and coordinating a group of researchers in the DARPA Control of Agent Based Systems (CoABS) program (members included: SRI, University of Rochester, University of

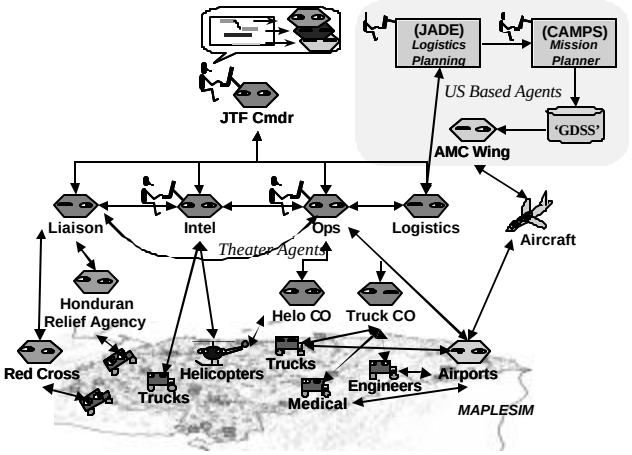

Figure 1. Agent Teams and Communication Flow

Oregon CIRL, CMU, OBJS Inc., Kestrel Institute, and Yale University) in developing a series of demonstrations of mixed-initiative agent-based computing in the context of large organizations. The working group and demonstrations were called MIATA, for Mixed-Initiative Agent Team Administration. The goal of these demonstrations is to explore and validate the potential for heterogeneous agent systems to support and function effectively *in collaboration with humans* within dynamic, distributed organizations.

Our main focus has been in a scenario with explicit roles for a number of users and software agents in a number of different offices, all coordinating to address a simulated disaster relief scenario based directly on the historical record of Hurricane Mitch, which devastated Honduras and the surrounding area in 1998. Hurricane Mitch struck Central America from October 26 to November 4 1998. It was the second most devastating hurricane ever recorded in the Western Hemisphere. 200 mph winds and 75 inches of rain left a death toll of over 11,000, with thousands more missing, and more than three million people left homeless or otherwise seriously affected. Honduras was affected most severely, with widespread flooding and devastation of its transportation infrastructure and villages.

The demonstration shows users directing and interacting with agents to:

- Form teams and assign tasks
- Gather intelligence about damage on the ground by interacting with a simulation (MapleSim) of the region.
- Plan for the deployment of relief supplies and repair equipment
- Manage logistical resources and distribution of supplies
- Respond to problems on the ground.

Agents communicate with users and other agents within the military organization, as well as agents representing non-governmental organizations like the Red Cross. Figure 1 shows a pictorial representation of the users, agents and the communication flow between them. Field agents act through a simulation of the ground state in Honduras as the hurricane passed through (MapleSim, provided by CMU). Users interacted with the organization at six different points, but primarily through either mixed-initiative tools performing specialized functions (logistics planning and scheduling tools developed in prior efforts and then 'agent-ized'), or through personal assistant agents specialized to support specific command roles. Our focus in this paper will be mostly on the latter.

The ground level simulator MapleSim models the actions of the hurricane as well as a large number of 'Field Agents', representing trucks, planes and helicopters and their occupants. The OMAR agent framework at BBN and the PRS system at SRI were the reactive plan execution systems used to model the agents representing the vehicles' drivers, as well as the engineers, and medical relief teams that moved about the country. All interactions with the MapleSim simulator are by KQML (Finin, Labrou, and Mayfield, 1997) messages, mostly transported over the CoABS GRID (GITI 2000), a mechanism designed to support communications between different agent frameworks. These messages enable the agents to change location, gather information, and change the state of the simulation world by performing a variety of basic relief tasks. Tasks modeled include the distribution of food, water and shelter, the providing of medical relief, and the repairing roads and bridges.

A second class of agents, Task Management Agents, also developed mostly in OMAR and PRS, schedule, task, and monitor other agents. Often these agents are also used as information collectors: collecting and disseminating information gathered by and for other agents. Typically, Task Management Agents manage field agents, but in some cases they manage other Task Management Agents. Examples of Task Management Agents in the disaster relief scenario include Helicopter and Truck Company Commander Agents, AMC Wing Commander Agents, Truck and Helicopter Scheduling Agents, and the Red Cross Relief Agent.

Finally, a third class of agents – really clusters of agents — are the Personal Assistant Agents (PAAs). These groups of agents work in a tightly coupled fashion to support mixed-initiative interactions between the agent organization and the human users. Work to date has focused primarily on three Personal Assistant Agent clusters: A cluster supporting the Joint Task Force (JTF) Commander, modeled using the TRIPS system from the University of Rochester, and PAAs supporting the JTF Intelligence (J-2) staff and Operations (J-3) staff. The latter two each manage their own teams of agents as well.

Team Formation and Tasking

One of the key attributes of the Hurricane Mitch scenario was that the organization that responded was a dynamically formed team, called a Joint Task Force. These teams are based on doctrine describing key team member roles and responsibilities, but they are not pre-existing organizations themselves. JTF teams are formed to address individual crisis situations.

In order to model flexibly formed mixed human/agent organizations, our agents use explicit models of the team organizations that they could be asked to participate in. Team models consist primarily of a set of potential team intentions (capabilities) and a set of roles for agents to play, describing the capabilities required and the lines of authority between agents filling their agreed-upon roles. Our objective was to model an interactive process between a designated commander and his/her PAA as they formed a JTF team, based on shared knowledge of this doctrine, and knowledge of the intentions relating to team formation in general.

The JTF Commander's PAA was developed by integrating OMAR agents with the TRIPS system (Ferguson and Allen 1998), from the University of Rochester. It consists of a cluster of agents presenting a multi-model interface for monitoring and managing agent team tasks, and a map-based tool for interacting over domain elements. The TRIPS system provided the spoken language capabilities for input and output, the interpretation of user directives, and the planning of responses. The integration was in many ways similar to that described in (Burstein, Ferguson and Allen, 2000) for the integration of TRIPS and the CAMPS scheduling system.

As part of our disaster relief scenario, the JTF Commander directs the formation of the JTF support staff through a spoken dialog with a PAA. As the dialog begins, the human commander queries the PAA to determine what other humans and their supporting PAAs are available to perform a number of critical roles such as Intelligence (J-2), Operations (J-3), and Logistics (J-4) as part the JTF team. An early portion of the interaction between the JTF Commander and his supporting PAA is shown in Figure 2. All dialog is spoken unless otherwise indicated.

Cmdr.: Establish a Joint Task Force at Soto Cano.
PAA: Alright
Cmdr.: Show me the officers there.
PAA: *<Displays a table of local officers>*
Cmdr.: Assign Captain Smith to J-2.
PAA: Ok.
Cmdr.: Major Jones can be the J-3.
PAA: Alright.
Cmdr.: And make Captain Brown the J-4.
PAA: Ok.
Cmdr.: Show me the team objectives.
PAA: *<Shows checklist of objectives for a JTF task force.>*
Cmdr.: *:<Selects objectives based on orders received>* Inform the team.
…

Figure 3: Team formation dialog

Intentional models of agent team formation and coordinated action have been described by a number of researchers (e.g., Cohen, Levesque, and Smith 1997; Grosz and Kraus 1996, 1999; Rich and Sidner 1997; Tambe and Zhang 2000).To support the dialog of figure 3, we needed to view the human commander's intention to form a team as itself a collaborative activity between the commander and a PAA. That is, mixed-initiative team formation and management, PAAs need a model of the protocols and intentions involved that is itself *a joint plan between the PAA and its user*. This shared plan is used by the PAA when interpreting user directives and for reasoning about how to support a human users intention to form or manage a team. Figure 3 shows some key early parts of the model we are developing of the intention/task hierarchy we associate with this process. The upper tasks are performed jointly, while expansions may have tasks primarily performed by one party or the other. For tasks like determining objectives, whose purpose is sharing information, the expansion may simply be a dialog to achieve a shared knowledge state. For others, like assigning agents to roles, specific tasks can be done by the PAA to support the process.

Team formation task models like that shown in the figure can be used to establish a local context in which to disambiguate user directives, and also reason about when it is appropriate for the PAA to perform a supporting task. For example, in the expansion of the Assign Roles intention, nodes shown in gray are those that the PAA can perform itself, in support of that joint intention. That is, it can find candidate agents to fill roles, and it can perform the inter-agent recruitment dialog between the commander+PAA and other human+PAAs. This model leaves to the user the job of making the final selections among candidates identified for roles.

The model is useful for disambiguating user directives in statements like 'Assign Captain Smith to J-2', which can be interpreted in this context as suggesting a role assignment, and hence result in the PAA initiating the role recruitment protocol. In an alternate context, the same assertion might be a directive to assign Captain Smith to be a member of the subordinate J-2 (intelligence) team.

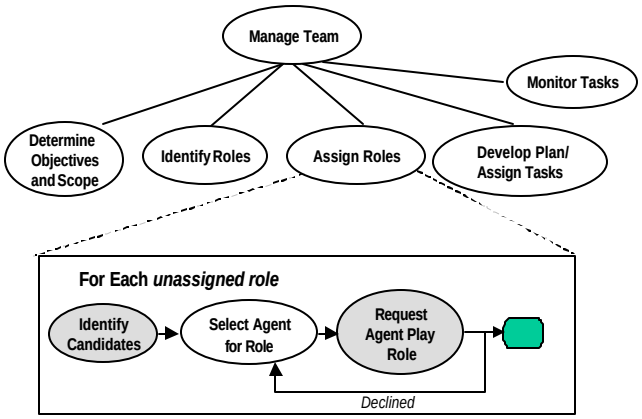

Figure 3: Team Management Tasks

Team Role Recruitment

The team formation task model leads the PAA to initiate the process of recruiting an agent to a team role whenever the user has specified a selection. The interactions between agents that start with a request to perform a tasks or accept a team role assignment require a potentially extended sequence of messages referred to as a *conversational policy* (e.g., Bradshaw, et al. 1997). A recipient agent's decision to reply by accepting or rejecting the role (or task) is a function of a number of factors, including:

- The agent having all of the capabilities for the role,
- The time availability of the agent

♦ The tasking agent's authority over the recipient agent [1]

For task requests, the following additional considerations also apply:

♦ Its prior intentions (or not) to be cooperative (as by having accepted role on the requester's team), and

♦ The existence of other conflicting tasks or intentions.

Accepting a specific role on a doctrinally defined team means accepting the joint intentions of the team as a whole, and establishing intentions to perform the set of tasks associated with the role, when required to do. The decision to accept a role is, therefore, dependent upon the agent's ability to perform all potential tasks associated with the role. If there are immediate tasks associated with the role, then any pre-existing intentions it had that may conflict with the responsibilities of the proposed would be justification for declining. In principle, although we do not address this in our current model, agents might also be able to accept a role 'with qualifications'; when they can perform only a subset of its described capabilities, or have other responsibilities taking some part of their time.

We believe that in a mixed human/agent environment, a requestee declining a request to accept a role (or, more generally, do a task) either because a task or resource conflict exists, or because it otherwise lacks the capability to perform the request, should report the reason(s) for declining. This is extremely important in a mixed-initiative environment, because it enables the tasking agent (frequently, with a user) to try to respond in a way that addresses the problem or resolves the resource conflict. When tasks mostly originate from a team leader or commander, who also has authority to shift resources, it is the availability of these rationales that enables interactive plan adaptation to occur.

As in other shared plan/shared intention models, once all key roles have been assigned and accepted, all team members are informed of the identities of the other team members, their roles, and the initial team objectives. This activity is triggered in our scenario by the 'Inform the team' command from the user.

Teams formation can also occur by the autonomous actions of a Task Management Agent. In our disaster relief scenario, while the upper echelon of the JTF was formed by interactions between the human JTF commander and PAA, supporting subteams were sometimes formed autonomously by that team's lead agent.

Status Reporting Policies

In our team framework, all assigned tasks have associated with them a default intention to report on task status when asked, or at completion. This intention is subject to modification with each request. Information about to whom (what agents) and how frequently to report task status accompanies tasking requests. Although, by default, task failures or completions are reported to the tasking agent, this needn't be the case for a variety of reasons. The tasker may have an associate agent that manages status information, or the tasker may have been an agent other than the team leader, in which case multiple reporting lines may be used, with potentially different levels of detail required. For example, the JTF Commander may request the J-2 (Intelligence officer and subteam) perform a survey of damage in a region and report the results to another team member, the operations officer (J-3). The J-3 would then be told if that task failed, but a full status or failure report with the reason for failure might only be sent to the Commander.

Agents may also do reasoning about whom to report on failures and delays. These decisions can depend greatly on knowledge of the team objectives, task dependencies, and such things as resource control or allocation. For example, the commander (a human) might direct the J-3 to 'loan' a helicopter to J-2 to support information gathering. Having done so, the J-2 could infer that the J-3, in addition to the commander, must be informed if that helicopter is damaged or delayed, because the resource was to be returned to the J-3's control for future use.

A related issue is how much to report about information collected or problems encountered. In a large hierarchical organization, it is desirable that only failures or delays in major assigned tasks be reported to supervisory agents or their users. Advice (Myers 1996) can be used to say when a situation should be reported immediately versus periodically, and in a detailed or summarized form, or not at all. This kind of customized reporting was developed initially in the TRAC system that implemented the J-2 team, and is described in Myers and Morley (2001).

Failure Reporting

We view it as critically important in mixed human/agent organizations that agents provide justifications when declining requests, and explanations when reporting current or anticipated failures or delays in tasks they have accepted. This is because the people and/or agents managing the team may be able to use this information to make better resource management and task allocation decisions to overcome the problems. As subordinate agents fail on tasks or exceed the expected time to complete assigned tasks, their manager may decide to give them more time, find additional resources to support them, reassign the tasks to other agents, perform tasks themselves, etc. Which of these things to do requires information about the failure. It does, however, place a burden on software agents to explain their reasons for concluding they have or will have failed – a requirement that is unusual in current-day agent systems.

A related issue is how failure information propagates up the chain of command in a hierarchical organization. In MIATA, only simple causal justifications could be reported from most low-level agents (e.g., delivery failed because a road was blocked). Task Management Agents reported failure when tasks they had been assigned by a higher level

[1] As a means of prioritizing tasks, only. We are assuming a basic cooperative stance by the agents.

agent failed because a component task failed (whether they intended to do the task themselves, or had intended a subordinate do it).

One might think that it would be sufficient that an agent report on failures only for the tasks it was assigned by others, but there are many cases where *patterns of failures* in low level subtasks can be worth reporting because they cumulatively indicate the need for replanning. An example here would be a mid-level Task Management Agent noticing that many of the trucks under its command were reporting failures to traverse roads in the same region due to flooding. This information, if discovered, could dramatically change the dispatcher's planning for future tasks in the same area.

Similar arguments apply for agents that do explicit task planning and constrained resource reasoning, such as schedulers. It can be quite difficult to come up with precise explanations (in the sense of specific, single causes) for the failure of a scheduler to schedule individual tasks among many. Individual tasks may be left unscheduled (or be scheduled late) due to the confluence of many constraints, such as shared resource bounds, and multiple tight deadlines. In a mixed-initiative environment where a commander is relying on subordinates to do scheduling, the most useful kinds of explanations to produce in such cases are those that enable the commander to replan strategically by making alternative resource allocation decisions, or selectively prioritizing or relaxing deadlines. Helpful 'explanations' therefore would include things such as indications of which resource bounds or time bounds were implicated in many tasks being delayed or failing to be scheduled. For example, a scheduler that fails in an attempt to replan after being told that ground agents cannot reach a town due to flooded roads should report both the failure and the condition that many roads were blocked. This assists the user to consider a different kind of resource (helicopter) to achieve the objective.

Cooperative information sharing

When a user is individually tasking the agents on a team, he or she will either have to be very explicit about the information passing expected between the agents, or else the agent team needs to develop appropriate information sharing strategies dynamically. Since it may be hard to predict in advance what information each agent will need, we saw it as important to address the problem of dynamic information sharing.

Put generally, if team agents intent to support other team agents' intentions, how do the agents become aware of what support they can provide? Must they be told of every goal that each other agent is pursuing? Or is there a policy by which they can be told when cooperation would be useful?

Consider the dialog in Figure 4 from the MIATA scenario.

Cmdr: Which are the hardest hit towns?

PAA: *<Brings up a map of Honduras and highlights towns.>*
Cmdr: Which towns are most populated?
PAA: *<Highlights another set of towns on the map.>*
Cmdr: Which towns haven't we reached yet?
PAA: *<Highlights another set of towns on the map.>*
Cmdr: *<Draws a region on the map.>* Let's collect intel in this area first.
PAA: Ok. *<Gives task to J-2.>*
Cmdr: *<Draws a region on the map.>* Provide relief in this area.
PAA: Ok. *<Gives task to J-3.>*

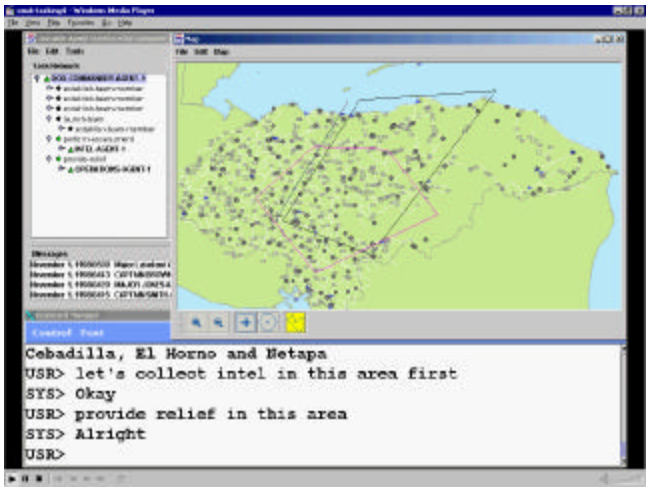

Figure 4. JTF Commander tasking team agents

After an initial survey is completed by the J-2, its partial findings are reported to the JTF commander's PAA, who in turn briefs the human JTF commander by displaying a summary of the damage levels and indicating which towns were found to have damage on a map. The commander's response is to direct each agent to begin their primary team tasks by focusing in what are now observed to be severely damaged regions. This directive causes the team agents to reprioritize which towns they will address first. For the J-2, this means completing its information collection task relative to those towns, while for the J-3, it means planning the delivery of goods to those towns.

The issue is how the J-3 is to become aware of the information being collected by the J-2 that it needs for its relief planning. In general, we would like these agents to become aware of their potential cooperative information sharing without the user having to direct that sharing.

To address this, we developed a model by which the agents can communicate about their information sharing needs, so that information support relationships can be formed dynamically without a tremendous amount of intent reasoning and situation awareness by each agent about the state of every other agent in the organization. It requires a fairly straightforward communications policy activated by the formation of new intentions by each agent.

Using the model, agents, when given specific tasks, can determine whom to tell about their status, failures, and information that they discover that is potentially needed by others on their team or in the larger organization. An important goal was that it be flexible enough so that users directing teams can initially make assumptions about agents' policies for information sharing, but change the information flow behavior dynamically. Details of the model summarized below are described in (Burstein and Diller, forthcoming).

The approach is based on the announcement by each agent of *information requirements* and anticipated future capabilities to do *information provision*. Expected information provision capabilities are based on intentions that can lead the agent to acquire information. These announcements to teammates establish the conditions for cooperative information sharing.

Previous teamwork models (e.g., Tambe and Zhang, 2000) have included preliminary general rules for some kinds of information sharing among team members, based directly on the goals of local teammates who share mutual team objectives and have partial shared plans. For example, in TEAMCORE there are generic rules for sharing information about observations that indicate the goals of other team agents have already been achieved or cannot be achieved. One can imagine a variety of other rules also supporting specific variations on the general *principle* that cooperating agents should provide the information needs of other agents based on their shared knowledge of the plans that those agents are pursuing. The difficulty is the amount of reasoning that may be required if the such policy rules are in a domain or context independent fashion, and the degree to which sharing of individual agents' plans is required to support those policies. This would be especially problematic in large organizations, where the agents would be overwhelmed by the reasoning required to anticipate the needs of large numbers of other agents. The possibility for this kind of reasoning being based directly on the full sharing of each others' plans is also limited in practice both by the variety of kinds of software agents likely to be found in an organization and by the need for interleaved action and planning in a continuous environment. Agents would need to reason not only about the information required as preconditions to other agents' plans, but the information needed when those agents were forming their plans in a particular context.

A simple approach to this problem, which works in some cases, is to make it an explicit stated objective of a one agent to acquire and provide particular kinds of information to a teammate as part of a team plan. This approach is, in a sense, the most direct way to engineer agent systems so that key pieces of information are available to the agents needing it. However, it is too brittle in several ways. First, the information requirement of one agent must have been explicitly anticipated by the developer of the plans of the other agent. Second, information needs may be conditional on circumstances, so it may be difficult to engineer reactive plans that anticipate when to acquire and provide the information at the right time, even if agents are sharing information about their intentions as they are formed.

This technique is also unworkable in larger organizations. The dispersal of information both within teams and also between teams is critical to large organizations. In the simulated MIATA organization, there are a number of heterogeneous teams (Intelligence, Operations, Logistics, Red Cross) that have a high degree of overlap in their information requirements. For example, field agents, such as Delivery Trucks and Engineers (who also use trucks) may be working in overlapping regions and desire to traverse many of the same roads and bridges, regardless of whether they're part of the Operations team, the Intelligence team, or as part of the Red Cross. Helicopter teams can often identify information of great utility to these trucks, and vice versa. Passability information is needed for all of these agents to plan routes. Furthermore, these agents, in traveling, may obtain information which, while not relevant to themselves, is of value to other agents who may or may not be directly a member of the same team.

Our model tries to strikes a middle ground, by adding a policy for explicit announcement of *anticipated information needs and provision capabilities* so agents who might have the ability to acquire the information, or by happenstance gain access to the information, can be aware of the others' need and signal a capability to provide it. The goal is promote sharing, to the extent possible, of the critical information needs and capabilities with both local team agents and those on remote teams within the larger organization.

The advertisement messages are triggered either by the acceptance of a role or when specific intentions are adopted by an agent in support of individual or teams goals:

Information Provision (IP) advertisements declare that the agent sending the message has an *intention to* achieve some purpose or execute some plan that results in it having information of the specified type. This signals an *intention that* it will answer queries with the identified classes of content, or provide such information on receipt of a subscription request. Information may be requested either on a periodic or 'as learned' basis. Information Provision advertisements are not retracted unless the original intention is aborted. *IP* advertisements are denoted as either *Active* (IP^A) or *Passive* (IP^P), depending upon whether the agent is actively pursuing the acquisition the information, or merely serving as a passive, but opportunistic gatherer and provider of that information.

Information Requirements (IR) advertisements declare that the agent sending the message has an *intention to* achieve some objective requiring the information. The requirement may be either for the purpose of planning how to achieve the intent, execute a conditional plan, or for use during execution (i.e., processing the information).

Simultaneously advertising an *IR* and *IP* over the same class of content suggests the agent's role as a 'knowledge

source' for that information in the future. The agent will both collect the information from anyone who provides it and provide the information when needed to others. Furthermore, some 'knowledge sources', like the J-2 in MIATA, are *active* sources, in the sense that they will perform actions to acquire the information (or direct teams that do so). Such agents issue IP^A advertisements, which further indicate that they may adapt existing plans acquire content they do not have at the time queried in order to acquire the information sooner.

IP and *IR* advertisements associated with an operation are immediately announced to team members upon the agent's intention to perform that operation. An agent, upon learning of an *IP* advertisement that matches the class of content it requires, may initiate a subscription protocol with the agent advertising the *IP*, enabling it to automatically receive the desired information upon its availability, or with a specified delivery schedule. The subscription request details the specific information desired, the rate at which the information should be delivered (e.g., on-occurrence, on-completion, or periodic with a given period, such as every once a day), and the level of detail to be provided (e.g., full vs. a specified summary level). Subscriptions may be removed after an information requirement ceases to exist.

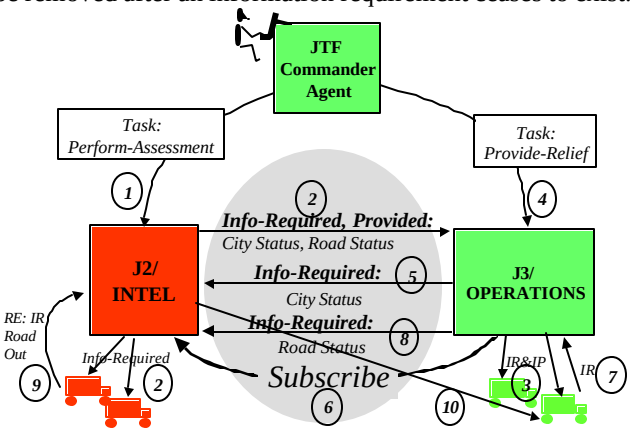

Figure 5. Information coordination protocol using *IP* and *IR* advertisements

Figure 5 illustrates this process. In MIATA, the teams are organized hierarchically, though that is not critical to our model. What is important is that agents are often members of more than one team. For example, the Operations (J-3) agent is a member of the JTF staff, but also is leader of the subordinate Operations team. These multi-team agents enable the sharing of information across teams. IP and IR messages are forwarded when received to all teammates on teams other than the team from which the message originated. In the figure, we see an example of the resulting message traffic. At time (*1*), the commander tasks the J-2 to survey the region. This causes an IP+IR message to all teammates on both the commander's team and the subordinate J-2 team (2). At time (*3*), the J-3 forwards the message to its subordinates. At time (4), the commander tasks the J-3, causing IR messages (5) to be sent out. Also, since J-3 already had received a similar message from the J-2, J-3 issues a subscription request for the survey information that is being collected (*6*). When the J-3 tasks its trucks to begin delivering goods, they issue IRs for the road information they each need for local planning (*7*). These messages are forwarded to the J-2 (8), who responds when the survey trucks find that information (*9,10*)

The information needed for this policy is part of the characterizations of agent roles. An agent role, $R(N, T, \rho, C, I, M, S)$, includes six main elements: the name of the role (*N*), the team of which this role is a member (*T*), the set of resources required by this role, the capabilities to be provided by agents filling the role (*C*), the default intentions initiated by accepting the role (*I*), the set of roles which are managers of agents in this role (*M*), and lastly, the set of subordinate roles to this role (*S*). *IP* and *IR* advertisements are explicit aspects of capabilities and intentions. Capabilities are things that role agents may be asked *intend to achieve*, $C(Sig,IP,IR,\rho,Rep)$. They are described by these elements:

- A message signature (*Sig*) for requesting the capability of the agent,
- Information provision ($IP(\pi)$) and information requirement ($IR(\pi)$) forms containing descriptions of the type of content provided or required as a semantic 'pattern' (π), which is further refined as specific intentions are formed for the capability, and the advertisements created.
- Required resources (ρ),and finally,
- A set of default reporting guidelines (*Rep*).

Reporting guidelines, $Rep(t,P)$ are default information subscription requests associated with a task. They include a description of the report type (status, failure, problem, result) and its information content, and a set of policies, $P(r,p,d)$ which includes the receiver (*r*), of the report, the reporting period (*p*), which is the time interval for reporting, but can also be on demand or upon the occurrence of an event, and lastly the level of detail (*d*), to be included in the report.

Conclusion

We have tried to illustrate some of the many roles that intentional reasoning can play in the coordination of dynamic team performance for organizations composed of heterogeneous collections of users and software agents. The discussion was based on our experience developing a large-scale demonstration of this kind of situation in collaboration with a six other agent research groups. The model was motivated by the relative complexity of the MIATA demonstration, containing over 100 agents and six users, developed by different groups, all supporting a relatively detailed simulation of a historically documented hurricane disaster relief scenario. Due to the scale of the development effort, many theoretical issues relating to the intent inferences described here are only partially worked

out. However, we feel that the work thus far has given us an important perspective on a number of the issues involved, and the kinds of reasoning that will be required of agents supporting such organizations.

The first part of our discussion focused on the role of task models describing team formation and management functions in the intent inferences resulting from the dialog between a human team leader and a personal assistant agent. Independent of task models within the domain of activity that the team itself will pursue, this generic task model will be needed to address the management of the team itself.

We discussed the role of status reporting by agents in supporting team leaders coordinating activity in dynamic, changing environments. In particular, we note the importance of explanation of failure as critical to effective decisionmaking by managers supervising agent teams.

We also presented a framework for dynamic information sharing among teams of humans and agents working as parts of large, frequently hierarchical, organizations. The key observation is that information needs, as well as agent intentions, must be shared among teammates and across organizations. In many cases, the information needs and provision capabilities of agents will require wider dissemination than the intentions of the agents. By using an approach where the announcement of intentions is accompanied by information about relevant information needs and expected information acquisition, we can create an environment where agents can develop their own information flow models dynamically. This is critical to effective workflow in mixed-initiative command and control environments as it removes the burden from users who might otherwise need to explicitly plan for and describe which agents needed to communicate with each other to achieve team objectives.

Acknowledgments. This work was funded under the DARPA Control of Agent-Based Systems program through contract F30602-98-C-0168.

References

Bradshaw,. J. M., Dutfield, S., Benoit, P., and Woolley, J. D. 1997. KAoS: Toward an Industrial-Strength Open Agent Architecture. In J. Bradshaw (Ed.), *Software Agents*. Cambridge, MA: MIT Press.

Burstein, M. H. and McDermott, D. 1996. Issues in the development of human-computer mixed-initiative planning systems, In B. Gorayska and J. Mey (Eds.) *Cognitive Technology*, Elsevier Science, pp. 285-303.

Burstein, M. H., Mulvehill, A. M. and Deutsch, S. 1998. An Approach to Mixed-Initiative Management of Heterogeneous Software Agent Teams. *In Proceedings of the 1998 Hawaii International Conference on the Systems Sciences*.

Burstein, M. H., Ferguson, G. and Allen, J. 2000. Integrating agent-based mixed-initiative control with an existing multi-agent planning system. In *Proceedings International Conference on Multi-Agent Systems*.

Burstein, M. H. and Diller, D. 2002. A Framework for Dynamic Information Flow in Mixed-Initiative Human/Agent Organizations. *Applied Intelligence*. Forthcoming.

Cohen, P. R., Levesque, H. R., and Smith, I. 1997. On Team Formation. In J. Hintikka, and R. Tuomela (Eds.), *Contemporary Action Theory*. Synthese.

Deutsch, S. E. 1998. Interdisciplinary foundations for multiple-task human performance modeling in OMAR. In *Proceedings of the 20th Annual Meeting of the Cognitive Science Society*, Madison, WI.

Ferguson, G. and Allen, J. 1998. TRIPS: An Integrated Intelligent Problem-Solving Assistant. In *Proceedings of AAAI-98. AAAI Press.*

Finin, T., Labrou, Y., and Mayfield, J. 1997. KQML as an agent communication language in Jeff Bradshaw (Ed.), *Software Agents*. Cambridge, MA: MIT Press.

Grosz, B. J. and Kraus, S. 1996. Collaborative plans for complex group action. *Artificial Intelligence*, 86, 269-357.

Grosz, B. J. and Kraus, S. 1999. The evolution of Shared Plans. In M. Wooldridge and A. Rao, *Foundations and Theories of Rational Agency*, pp. 227-262.

GITI/ISX Grid Vision Team, "The CoABS Grid: Technical Vision," http://coabs.gobalinfotek.com, 2000.

Myers, K. L. 1996. Advisable Planning Systems. In A. Tate (Ed.), *Advanced Planning Technology*. Menlo Park, CA: AAAI Press.

Myers, K. L. and Morley, D. N. 2001 Directing agent communities: An initial framework. In *Proceedings of the IJCAI-01 Workshop on Autonomy, Delegation and Control: Integrating with Autonomous Agents*.

Norman, D. A. 1997. How People Might Interact with Agents. In J. Bradshaw (Ed.), *Software Agents*. Cambridge, MA: MIT Press.

Rich, C. and Sidner, C. 1997. COLLAGEN: When Agents Collaborate with People. In *Proceedings of the International Conference on Autonomous Agents (Agents '97)*.

Tambe, M. 1997. Towards flexible teamwork. *Journal of Artificial Intelligence Research*, 7, 83-124.

Tambe, M., and Zhang, W. 2000. Towards Flexible Teamwork in Persistent Teams: Extended Report. *Autonomous Agents and Multi-agent Systems*, *3*, 159-183.