Applied Intelligence 20, 283–298, 2004

A Framework for Dynamic Information Flow in Mixed-Initiative Human/Agent Organizations

MARK H. BURSTEIN AND DAVID E. DILLER
BBN Technologies, 10 Moulton Street, Cambridge, MA 02138, USA
burstein@bbn.com
ddiller@bbn.com

Abstract. We are interested in developing models of and support for mixed-initiative human control of software agent teams, especially in the larger context of dynamic, real world organizations. In this paper, we describe a model for the establishment of cooperative information sharing among agents on teams formed dynamically for particular purposes within such organizations. We argue that effective information sharing in the presence of such teams requires the active dissemination of descriptions of current and future information needs to both local teammates and to the larger organization. Only by this mechanism can one avoid having to make explicit at design time who will provide each bit of the information. We consider how information sharing within the organization can be promoted not only for the immediate goals shared by a tightly coordinated team, but some of the likely information needs of the larger organization going forward. We illustrate the model by describing its application to a large-scale agent-based simulation of the US Military's disaster relief response to the devastation caused in Central America by Hurricane Mitch in 1998. The demonstration was developed in conjunction with a large group of researchers representing eight different institutions.

Keywords: agent interoperability, mixed initiative, multi-agent systems, information sharing, agent coordination

Introduction

We are in an era where the informational and computational requirements of organizations such as the US Military, NASA, the FAA, and most large corporations require a synergistic coupling between humans and software systems. Indeed, many traditional and web-based distributed systems now cross organizational boundaries. Of critical importance to the ultimate success of such systems are the interactions between humans and the software systems playing roles in the overall distributed environment. As in large purely human organizations, increasing the number of agents within the organization increases the time and effort required by human managers to manage the agents doing the work. As every web user knows, agents, like search engines, can generate tremendous amounts of information, both useful and not. Furthermore, as many people (e.g, [1]) have pointed out, human users of agent systems are unlikely to develop trust in the capabilities of the agents until the agents can be employed much like the human subordinates, in a way that is not intrusive, and can be carefully monitored. Only when agents can respond effectively to task directives, collaborate independently with other agents within the organization, and handle a variety of failure conditions appropriately will agents be an effective addition to the organization. Our goal is to design mixed-initiative command and control strategies that will increase the flexibility and robustness of multi-agent systems without substantially increasing perceived complexity to human managers within the organization.

For several years, we have been developing mixed-initiative collaborative tools to support humans in organizations composed of humans and software agent systems [2, 3, 4]. We take the view that the role of the humans in this kind of environment is that of 'team managers', tasking subordinate human/agent teams to work together toward shared goals, helping establish coordination and information sharing policies, and providing advice on priorities and resource utilization. To assist in these tasks, the personal assistant agents (PAAs) supporting human users must receive and relay concise and timely reports on both environmental (domain) conditions and task status. Since the humans are also working as part of the teams, they need to be able to engage in dialogs with their agents about the type and volume of information they are willing to receive, when they can be interrupted, etc.

Most organizational objectives *extend over long periods of time*, so directives to agents playing ongoing roles in team plans must lead to the establishment of ongoing information support relationships. We see this as akin to the dynamic formation of workflow models. This paper describes one possible approach for team agents to establish such relationships under the guidance of a human team manager, based on a combination of organizational doctrine and the specific taskings they receive. We see this approach as integrating and extend notions from information flow modeling found in AI-related workflow models [5, 6] which also integrally contain both humans and software 'agents', and models of execution for continuous domains [7].

This paper focuses primarily on the information flow issues surrounding this vision. When a user tasks a team of agents, the agent team needs to develop appropriate information sharing strategies. If agent intentions within a team extend over time, and must support other agents' intentions, they must become aware of what support they can provide, sometimes before all team plans are in place. We propose a model by which agents can communicate about their information sharing needs, so that support relationships can be structured to support not only the immediate goals shared by a tightly coordinated team, but the likely information needs of the larger organization going forward. This means, when possible, discovering the future needs of functional sub-teams within the organization, including agents yet to be tasked in those subordinate teams, but whose expected *roles* make clear the kind of information needed.

The model proposed is based on representing and disseminating knowledge about information needs and information provision capabilities of agents. These agents are assumed to exist as part of a dynamic organization composed of interrelated heterogeneous agent teams and humans supported by agents. Using this 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 of the model is 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.

A key element of the approach is the announcement by each agent of *information requirements* and anticipated future capabilities to do *information provision*, for the kinds of information the agent may come to have as a result of its *intentions.* These announcements to teammates establish the conditions for cooperative information sharing. Furthermore, by associating information policies with default intentions of team roles and with the capabilities associated with roles that may be activated by team plans, agents can quite easily initiate dialogs to coordinate information sharing when accepting roles or initiating new tasks.

Previous teamwork models (e.g., [8]) 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. An example here is the set of the generic rules in TEAMCORE 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 implemented in particular agents supporting specific agent communities embodying 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 difficulties lie in the amount of reasoning that may be required (or possible) if the rules are stated only generically, in a domain and context independent fashion, and the degree to which sharing of individual agents' plans is required (especially for larger organizations). The possibility for full shared plans is limited in practice both by the nature of the agents in a heterogeneous environment 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.

One can also imagine a simple approach within the general outlines of the shared plans paradigm [9, 10] where it is a 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 another agent, which does not allow for dynamic cooperation. Second, information needs may be conditional on circumstances, so it can 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.

Our model tries to strikes a middle ground, by adding explicit interagent sharing of anticipated needs so agents who might gain access to the information can be aware of it 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 model includes a characterization of agent capabilities for team roles that makes possible the explicit designation of agents whose role in a team includes accumulating, distilling and regularly disseminating information. Since overall team purposes extend over time, periodic and 'as needed' dissemination of information can be expected to be an important function of some subset of the members of the larger agent organization. This critically includes information about failures and potential causes of future failures related to things such as resource limitations of the organization.

The MIATA Simulation Environment

We implemented our model as part of a series of simulation-based demonstrations by the MIATA (Mixed-Initiative Agent Team Administration) working group within the DARPA Control of Agent-Based Systems program. For the past two years, we have lead the MIATA working group[1], whose goal was to explore and validate the potential for heterogeneous agent systems to support and function effectively within dynamic, distributed human organizations. We are convinced that it is only because we developed a large organizational simulation model operating in a realistic scenario that we were able to understand and address the issues discussed here. Thus, we will frequently illustrate the issues discussed with examples from MIATA.

MIATA's main focus has been in a scenario where users and software agents were teamed in a number of different offices representing the Joint Task Force (JTF) that coordinated the US Military's response to the 1998 Hurricane Mitch disaster which devastated Honduras and much of the surrounding region. We built this scenario based directly on the historical record of the events from a number of sources, and information we were able to obtain from the government about the plans that were made by the US to airlift supplies to the region.

The MIATA demonstration had six human users directing and interacting with over one hundred software agents to:

- Form teams and assign tasks,
- Gather intelligence about damage on the ground by interacting with a simulation, (MapleSim) of the region during and after the hurricane passed through,
- Plan for the deployment of relief supplies and the repair of roads and bridges,
- Manage logistical resources and the distribution of supplies,
- Report and respond to problems 'in the field'.

The software agents used were a mix of 'agent-wrapped' versions of pre-existing software tools and agents developed specifically for the demonstration. The 'wrapped' agents were logistics planning and scheduling systems[2] of various kinds with graphical user interfaces developed to address the needs of users with military logistic planning tasks. The other agents in the model were developed in OMAR [13] and PRS [14], two very similar reactive procedural execution systems. They were mostly representing field agents (trucks, helicopters, aircraft) and their local commanders (truck, helicopter, and airlift wing company commanders, other staff agents on teams). We used several different agent communications mechanisms for different clusters of agents, which were made to interoperate via the CoABS GRID [15], an agent interoperation framework defined on top of Jini [16].

Our simulation of the US Military's disaster relief effort in response to Hurricane Mitch consisted of a hierarchical set of teams with humans (with agent assistants) leading the highest level teams. Figure 1 shows a pictorial representation of the users, agents and the communication flow between them. The main body of the team is a Joint Task Force (JTF), a dynamically formed organization structured by doctrine with a set of roles for different sub organizations. The

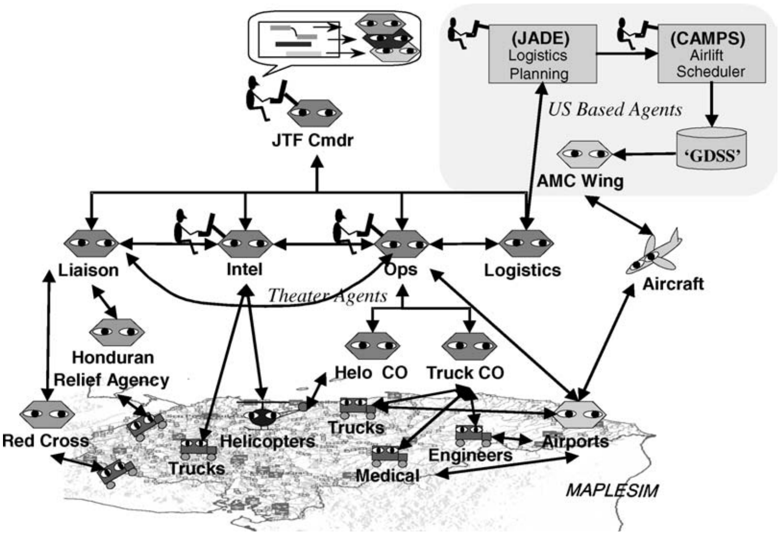


Figure 1. Agent teams and communication flow.

main teams reporting to the commander are an intelligence collection group (JTF-J2), an operations group (JTF-J3) and a logistics group (JTF-J4). The task force communicates with other elements of the larger US Military organization. The JTF Commander reports to a US Joint Commander (not shown), whose role in the simulation is to give the initial guidance to form the team, and to receive progress reports. The JTF logistics team works with the logistics arm of the Joint command and, indirectly, with the US Transportation Command, which creates schedules for the airlift of supplies that are executed by the wing teams. We also model interactions with non-governmental organizations like the Red Cross, which was also providing medical relief. At the base of the simulation, we have the 'mobile' field agents that act through messages sent to the MapleSim simulation of the ground state in Honduras. MapleSim, developed at CMU, models the hurricane passing through the country, and probabilistically creates the damage to be repaired. It also handles queries to towns for state information, but limits communication if there is (simulated) damage to communications infrastructure.

Collectively, the Joint Task Force team of humans and agents forms a hierarchical organization that interact with the simulation to perform a variety of basic relief tasks such as the distributing food, water and shelter, providing medical relief, and repairing infrastructure such as roads and bridges. At the lowest level, we simulated a large number of field agents, such as drivers of trucks and helicopters, engineers, and medical relief teams. These agents move about transporting cargo, gathering information and generating field reports for their team leaders. Task Management Agents (TMA), schedule, task, and monitor other agents. They are the leaders of the field agent teams. Often these agents are also used as 'information collectors'; agents that collect and disseminate (on request or periodically) information about field conditions for other agents on their teams. Personal Assistant Agents (PAA), support mixed-initiative interactions between agents and humans.

Interactive Team Formation and Tasking

In order to develop mixed human and agent teams that the humans, as team leaders, can understand, we support teaming models which can be represented as doctrinally defined parts of the larger human organization. These team structures, much like many kinds of athletic teams, can then be defined to have a set of roles with specific responsibilities related to the different team objectives (plays) the team can perform. Roles are defined to have a minimum set of required capabilities for the agents filling those roles, specified lines of authority for tasking, and default ways of reporting information back (to tasker, and to team leader). By describing communications paths in terms of these roles, agents, when recruited, can often be thought of as stepping into a workflow model with understandable information pathways.

This is somewhat unlike the TEAMCORE model described by Tambe and Zhang [17], in part because our model is geared to heterogeneous teams composed of mixtures of special purpose agents (software systems) that are directed by humans. These team leaders can do some of the retasking that the agents cannot do alone. We do pattern most of our model of team formation policies on TEAMCORE, and as such are building on the notions of agent team intent described by Cohen [18, 19] and in Grosz and Kraus' [9, 10] Shared Plans models, especially in the assumptions governing the conversational policies [20] used for establishing teams and team goals. The difference is that the process is user directed.

Cooperative Information Sharing

To form a team, agents are identified by the team leader and asked by that user's assistant to accept a team *role*, described in terms of a set of capabilities (tasks that may be assigned to that agent) and a set of shared objectives that the team as a whole may be tasked to achieve. Partial shared plans for each possible team objective describe generically the objectives that will be assigned to each team role player when a team objective is adopted at the direction of the team leader. The data flow model associated with each role in the team's plan is specified in a initial set of *information sharing and information needs declarations*, describing for each role the possible objectives requiring information and the information that they might provide when tasked. Other information sharing policies, universally specified for agents deal with their need to produce reports on task status and failures. Reporting requirements are included for the status of tasks, and the information gathered by an agent with a default policies governing the frequency and level of detail of such reports, and identifying the team members (roles) who should receive reports of particular kinds.

By accepting a team role, an agent (or agent with associated user) accepts the authority of the team manager to task them to perform the services defined for their role. The role of *team leader* on a team can be filled either by a software agent or a personal assistant agent (or cluster of agents) acting as a proxy for and in collaboration with a human. The team leader is responsible for the 'hiring' and initially tasking the team, which is based on objectives the leader receives from a higher authority. In the MIATA simulation, this is exemplified by the role of the Joint Task Force (JTF) Commander, a doctrinally defined position as a team leader of a JTF team.

Teams are formed through mixed-initiative dialogs between humans and PAAs in some cases, and by automated Task Management Agents (for subsidiary teams) in other cases. The primary JTF team is formed through verbal communication between the human JTF Commander and his PAA (see Fig. 2). The PAA was developed in collaboration with the University of Rochester, and utilized a TRIPS [21] agent cluster to manage the dialog with the user. Sub-teams are formed by JTF staff agents as required to support the tasks they accept as part of their roles in 'parent' teams. The team leader's assistant requests other agents to join a team through a sequence of messages embodied in a *communications policy* [20]. Agents asked to perform a particular role within a team, have the opportunity to either accept or reject the role, dependent upon the agent's ability to perform the role, as a function of their existing capabilities, and any pre-existing roles or tasks that may conflict with the proposed role.

***Human Commander**:* Establish a Joint Task Force at Soto Cano. Area of operations is Honduras.
***PAA Agent**:* Alright
Cmdr: Show me the officers there.
PAA: *<Displays a table of local officers and their areas of expertise. >*
Cmdr: Assign Captain Smith to be the J2.
PAA: Ok. *<Sends message to Smith's PAA>*
. . . Cmdr Selects other human/agent teams for other roles . . .
Cmdr: Show me the team objectives.
PAA: *<Shows a list of objectives defined as capabilities of this kind of JTF team.>*
Cmdr: *<Selects objectives 'provide relief' and 'repair infrastructure' from GUI>*
Cmdr: Inform the team.
PAA: *<Sends team objectives and initial shared partial plan to all teammates.>*

Figure 2. Team formation dialog.

Once a set of agents has accepted their team roles, the team leader (human + PAA) can begin describing specific team objectives (e.g., 'provide relief', including the delivery of food, water, and medical assistance). In Fig. 2, the last two lines of the dialog initiate this interchange between the JTF team leader and the staff agents. Each team role player is informed of the team objective(s) and their own initial objectives relative to those goals. These objectives may later be refined by providing additional directives or advice [22] to specific team members.

The JTF team objectives given by the commander in this dialog are to provide relief to towns in the region and repair the country's infrastructure, which we represent by repairing roads and bridges. The team plan for provide-relief calls for the J2 (intelligence team) to begin by doing a survey of the towns for their material needs. In support of the repair-infrastructure task, that team simultaneously tries to discover what roads and bridges are damaged. The result of this survey is provided to the human JTF commander via his or her PAA, to enable the commander to further elaborate, refine and prioritize the tasks of each primary team member.

When tasked, the J2 announces to the JTF team that it has an *intention to* provide the classes of information contained in the survey so that other agents can subscribe to that information as it is produced. This announcement eventually triggers teammates needing the information to engage in a dialog with J2 to establish information coordination policies based on different kinds of subscription. This kind of announcement is key to our dynamic information-sharing model. We call it an *information provision advertisement*.

Meanwhile, the J3 (operations) team leader, another human + PAA agent built in OMAR, was given the task of delivering the needed supplies to towns that are discovered to need it. It first must find out which towns those are and how much they need, and so it independently announces this information requirement to its teammates. If the J2 had not announced that it could provide that information, but had the information available, this *information requirement advertisement* would trigger a message from J2 to J3 stating that it could provide it.

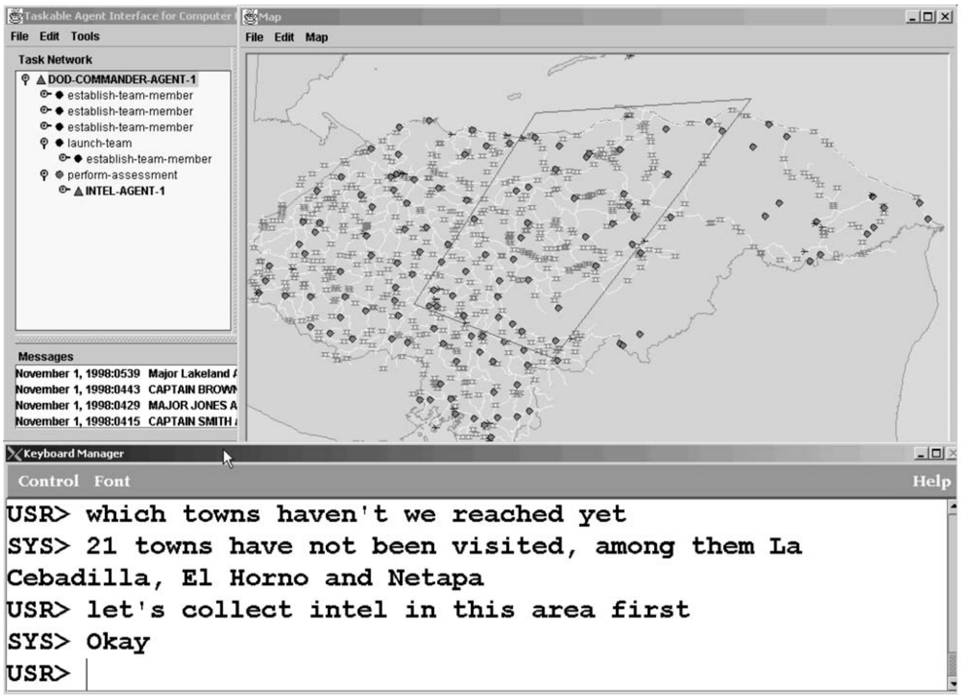
When tasked, the J2 also announces its 'interests' to its direct teammates and subordinates. This allows the J2 to get the cooperation of the whole organization in collecting the information. By immediately announcing its interest in collecting information on particular topics, it solicits help in this process from other agents who may opportunistically discover relevant information as they move about.

Associated with the J2 role's primary task in support of the JTF's provide-relief team objective is a description of its informational role in that overall team plan. This description includes a representation of the kinds of information it will be capable of providing to other team members to support their roles in the plan, and a corresponding description of the kinds of information it wants to collect so that it can fulfill that role. In terms of an AI blackboard systems metaphor, the J2 is a *knowledge source*, but it is also an *active* collector of information, through its direction of a sub-team of field agents that moves about to gather information. In MIATA, the J2 role was filled by an agent developed at SRI using PRS [14]. This agent also acted as a PAA for a J2 user, who advised the team in the field, and addressed failures and problems encountered by those agents. The overall J2 role also included the production of 'daily' summary information reports for the JTF commander.

Figure 3 illustrates an interaction between the JTF commander and the JTF commander's PAA regarding the results of the initial survey by the J2. After the initial survey is completed, its (very partial) findings are reported to the JTF commander's PAA, who in turn briefs the human JTF commander, by displaying a summary

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 J2 to commence Intel operations in designated region.>*
Cmdr: *<Draws a region on the map.>* Provide relief in this area.
PAA: Ok. *<Gives task to J3 to commence planning for cargo deliveries to the region.>*
…

Figure 3A. JTF Commander's reprioritization of agent tasking.



```
November 1, 1998:0539  Major Lakeland
November 1, 1998:0443  CAPTAIN BROWN
November 1, 1998:0429  MAJOR JONES A
November 1, 1998:0415  CAPTAIN SMITH
```

```
USR> which towns haven't we reached yet
SYS> 21 towns have not been visited, among them La
Cebadilla, El Horno and Netapa
USR> let's collect intel in this area first
SYS> Okay
USR>
```

Figure 3B. The JTF Commander's interface during task reprioritization.

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 core team task by focusing on what is now observed to be severely damaged regions. This directive causes the team agents to reprioritize (replan) which towns they will address first. For the J2, this means completing its information collection task relative to those towns, while for the J3, it means planning the delivery of goods to those towns.

Information Needs Registration and Provision within Agent Organizations

A key issue for human and mixed human/agent organizations are the policies for information flow to support the varied tasks of individual agents and groups within those organizations. When considering how agent teams might support such policies from the perspective of shared intention and shared plan theories, there are several difficulties. First, while these models provide some of the theoretical basis for motivating information flow, in terms of cooperation rules or policies to support the objectives of teammates, it may be difficult or impossible to implement these abstract policies directly in software agents with limited inferential capabilities, as is frequently the case in heterogeneous agent systems. So, for example, in Tambe and Zhang's TEAMCORE model [17], there are team member policies that will cause an agent A to inform team member B that the condition P that B was intending to achieve was observed, or had become unachievable. In the latter case, the observation required might well depend on A knowing (and being able to reason about) the full details of B's plan to achieve P, in order for A to determine that some precondition to a step in that plan was false or not itself achievable. A similar argument can be made for a policy to provide information to agent B when the information is required for B to develop a plan to achieve P. A would need to know *how* B would plan to go about achieving P before it could infer what B needed to know.

Second, the informational needs of agents can be arbitrarily context specific, which means that all team agents must know more about each other's detailed plans than even a Full Shared Plans model would require. Full Shared Plans [9, 10] allow for the case that some step r_i in the recipe R to achieve P are to be carried out by a subgroup of the team, and that all team members believe that there is a plan for that subgroup to achieve r_i, but they do not know what it is. In an environment where agents' planning and acting are interleaved, it is in general impossible for an agent to be certain that any particular piece of information will be useful for another agent's plan, or for them to develop such a plan.

This being said, there are many ways agents might determine that a piece of information *might* be useful, and cooperative behavior policies can be built on these methods. Consider the apparently simple case of an agent representing a truck that, as part of a shared team plan, is intending to deliver some cargo to a town. The *capabilities* of truck T are limited, and might be expressed simply as:

(carry T <*cargo*> <*loc1*> <*loc2*>)

and it might have a standard plan to achieve such objectives involving picking up its *cargo* at *loc1* and moving over a sequence of road segments r_i to reach *loc2,* where it unloads the cargo. When given a task, truck T would develop a plan by computing a sequence of passable road segments to traverse. If team agent A knew that T intended to deliver a cargo from $L1$ to $L2$, what useful information might it provide? Some possible things it might tell T about road conditions to assist in its planning are:

1. Which roads it knew were impassable *anywhere*,
2. Which roads it knew were impassable anywhere in a region containing $L1$ and/or $L2$,
3. Which roads it knew were impassable on a path A itself computed to go from $L1$ to $L2$,
4. Which roads it knew were impassable along the path T *announced it intended to take.*

While there is no absolute right answer here, it seems clear that 1 might be too much information, depending on the extent of A's knowledge, and 3 and 4 are not enough. In fact, option 3 requires that A plan without using all of its own knowledge of where roads were impassable, and still doesn't provide T with enough information for T to find a successful plan, just to reject one. Option 4 does much the same thing, while leading to a two-agent generate-and-test strategy. Option 2 seems the most 'helpful', since it is circumscribed to include mostly relevant information, and likely provides T with enough information to develop its own plan, while simultaneously not requiring a great deal of reasoning on A's part to gather that information.

So how is A to know that option 2 is the right information to provide? Without detailed knowledge of T's route planning process, it needs to know at least about the preconditions of the operators that T is using to plan the task, namely that the operator that is executed (traverse T <route-segment> <direction>) requires (passable <route-segment>), and that, due to the nature of the 2D space the segments are laid out on, the best paths are all in the region around the line segment connecting $L1$ and $L2$. Even this is likely to be beyond the reasoning capabilities of simple BDI agents who are themselves *programmed* to accomplish some specific tasks.

Our working assumption is that the agents likely to inhabit large mixed-initiative organizational systems are a heterogeneous collections of 'wrapped' versions of special purpose software systems, together with some more general purpose reactive execution BDI agents that assume the bulk of the 'middle agent' duties. This is essentially what we had in the MIATA system. Both PRS and OMAR were used in MIATA to implement the JTF staff agents, and the bulk of the agents that 'drive' vehicles in the simulation. They are very similar reactive procedural execution models with explicit representations of agent intent, and supporting parallel and conditional execution. They both have some capability for forward and backward inference, and can thus be made to handle a variety of reactions to observations (messages) outside of their primary tasks, and have some capability to respond to queries on their internal work status while working. These agents were supported by more special purpose agents for such things as detailed scheduling (for aircraft, trucks, helicopters) with specific information needs, and data repositories with distinct information provision capabilities.

If this mix of agent capabilities is typical of what one might use in a variety of real-world applications, then we would argue that the determination what information a teammate agent will need to achieve its announced intentions should, wherever possible, be done by a combination of the following:

- making a characterization of that information need explicit in the characterization of that agent's capability to intend that result, announced at team formation,
- having the agent announce a specific information need when it commits to achieve that objective.

The key observation here is that information needs, like the capability to provide information that we will address shortly, should be made explicit, rather than left to shared knowledge of planning operators. It is frequently assumed that a capabilities description for an agent, much like APIs in more traditional distributed object systems, are composed primarily of the set of message patterns that an agent will respond to, and perhaps a characterization of the response. Yet stating that an agent can respond to a query containing a variablized WFF provides almost no information, and declaring that a truck agent can respond to a request to perform (carry <*truck*> <*cargo*> <*loc1*> <*loc2*>) does not address our problem.

Characterizing Information Provision and Collection Information Requirements

To address this problem, we identify two kinds of communication about information to be advertised by all agents in an organization. These advertisements are associated with either the basic intentions associated with acceptance of a role or with specific intentions adopted by an agent in support of individual or team 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 to others when needed. Furthermore, some 'knowledge sources', like the J2 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 to acquire content they do not have at the time queried in order to acquire the information sooner.

IP and *IR* advertisements associated with operations are announced to team members as agents form intentions to perform them. An agent, upon learning of an *IP* advertisement that matches the class of content it requires, may initiate a subscription to 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 4 illustrates this process. At time t_0, agents B and C receive a team objective g from agent A, with subgoals g_1 and g_2, for B and C respectively. After developing intentions to perform for g_1 and g_2, B and C broadcast the *IP*s and *IR*s associated with those intentions. Agent B broadcasts an information requirement advertisement, IR_i requesting information of content class i and agent C broadcasts an information provision advertisement, IP_i^A, announcing its intent to actively acquire and provide information with content i. Both advertisements are broadcast to the team at time t_1. Agent B, upon notification that Agent C intends to actively acquire and provide content i, initiates a subscription protocol, $S(i,\phi,\delta)$, at time t_3, requesting information content i with frequency ϕ and level of detail δ.

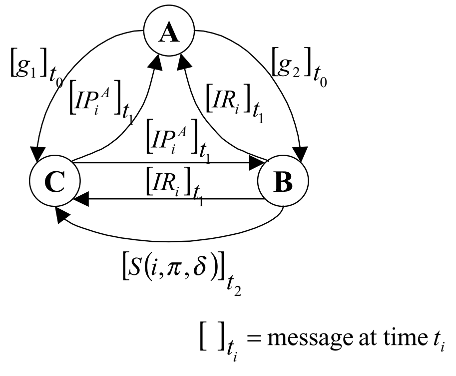


Figure 4. Information coordination protocol using *IP* and *IR* advertisements.

The information needed for this policy is embeded in 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 team 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 provided by 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.

The dispersal of information both within teams and also between teams is critical to large organizations. In MIATA, 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.

Information Sharing Protocol

In order to support information sharing through *IP* and *IR* advertisements, an information sharing protocol has been established to facilitate the dispersal of advertisements within and among teams. In MIATA, these 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 (J3) 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. A Liaison agent may be used, often in conjunction with a Translation agent, to facilitate communication and information sharing between organizations with no common members. In MIATA, the Red Cross and the JTF teams communicate via such an agent. We have developed a number of rules that govern dissemination of information advertisements between team members, and their forwarding to adjacent teams. A subset of these are shown in Fig. 5.

Rules 1 and 2 support the sharing of *IP*s and *IR*s associated with basic intentions that are a part of the acceptance of a role. Upon acceptance of a role, an agent shares all *IP*s and *IR*s associated with that role with all team members. Rules 4 and 5 govern the sharing of *IP*s and *IR*s associated with an agent's capabilities. Upon acquiring an *intention to* described by an agent capability, the agent shares the associated *IP*s and *IR*s with its team members. Rule 6 specifies that *IP*s and *IR*s should be 'passed on' to other team agents if the agent or team is not the originator of the advertisement. Rule 8 governs whom an agent should inform if and when it acquires information pertinent to an *IR*. Finally, rules 3 and 7 allow for the retraction of *IR*s upon completion of an intention. *IP*s are not normally retracted, as agent may continue to provide information. Additional rules (not shown) are required for determining when and how *IR*s should be combined. Ultimately, organizational workflow analyses may be needed to discover when it is advantageous for an agent to become an information collector/provider for a team of agents with intersecting information requirements.

```
(role :name truck-company-commander  :team J3-staff
  :capabilities
    ((task :signature (repair-objects :objects (?object (location ?object? ?region)))
         :info-provided () :info-required ()
         :required-resources (engineer-agent)
         :reporting
           ((report  :type status :receivers ((?tasker :period on-demand :detail full)))
             (report  :type failure :receivers ((?tasker  :period on-occurrence :detail full)))
             (report :type problem :receivers ((?tasker :period on-occurrence :detail full)))
             (report :type result :receivers ((?tasker :period 12 :detail full)))
     (task :signature (distribute-supplies :cargo-sources ?sources
                                           :deliveries ?deliveries)
           :required-resources (truck-agent)
           :reporting (...)))
 :managed-by (j3-agent))

(role :name truck-company-commander :team truck-company-23
  :capabilities ((task :signature dispatch-truck :from ?source :to ?dest :cargo ?cargo)
                 (task :signature deliver :sources ?sources  :deliveries ?deliveries))
  :role-intentions
    ((goal :signature (provide-information-support :to-role truck-agent)
                   :info-provided ((and (bridge ?id) (state ?id ?state) (region ?id ?region))
                                   (and (road ?id) (state ?id ?state) (region ?id ?region)) )
                   :info-required ((and (bridge ?id) (state ?id ?state) (region ?id ?region))
                                   (and (road ?id) (state ?id ?state) (region ?id ?region)) )
                   ))
    :manages (truck-agent scheduler-agent))
```

Figure 5. *IP* and *IR* dissemination rules.

Figure 6 illustrates, using a lisp-like syntax, the role of a Truck Company Commander in the MIATA scenario. The Truck Company Commander (TCC) role is a part of the operations (J3) staff and requires truck agent resources in order to properly perform the role. The TCC also *manages* a company of Truck and Engineer agents. This role includes the scheduling, (with the assistance of a scheduling agent) and tasking of its subordinates to perform individual tasks from the collective tasks it receives. Because a prime characteristic of a truck company involves the traversal of roads by Truck and Engineering agents, it is a responsibility of the TCC as team leader to collect and disseminate information pertaining to the passability of those roads. Thus, the TCC team leader role has a standard intention to *provide-information-support* which contains both *IP* and *IR* advertisements for the state of bridges and road segments. This enables the commander to collect from his team and others who may discover it which routes are impassable, and provide that information to the trucks being managed when they require it to plan routes.

Figure 7 shows the dissemination of *IP*s and *IR*s that enable the truck company commander to collect this route information from both the trucks managed and, importantly, from other field agents throughout the organization. Four teams (JTF-Staff, J2-Staff, J3-Staff, Truck-Company) are illustrated with circles surrounding the members of each team. Like the example illustrated in Fig. 4, the JTF Commander sends objective g, with subgoals g_1 and g_2, to the J2 and J3, triggering damage assessment and relief provision intentions, respectively, in support of g. The advertisements associated with these intentions are broadcast to all members of each team of which that agent is a member. For example, at time T1, the TCC broadcasts an *IP* and *IR* advertisement to the J3 Agent, the Helicopter Company Commander (HCC) and Truck agents #1 and #2. The information requirement IR_j^{TCC} represents the desire to acquire information content j, representing the state of bridges and roads within its area of interest. The information provision advertisement, IP_j^P, represents the passive acquisition and provision of this same information, enabling the TCC to be a source of road and bridge possibility for its Truck agents.

Using the rules described in Fig. 6, these same advertisements are passed between teams of agents. At time T2, the J3 informs the J2, who at time T3 informs

Rule 1. An agent (x), after accepting a role (R) on a team, tells all teams, of which it is a member, the information it can provide (*IP*s) as a function of its default intensions (I).

$$\forall x\ [\forall k \ni x \in T_k\ [\forall y \neq x \in T_k \supset \mathrm{tell}(x, y, IP_I)]]$$

Rule 2. An agent (x), after accepting a role (R) on a team, tells all teams, of which it is a member, of any information requirements (*IR)* associated with its default intentions (I).

$$\forall x\ [\forall k \ni x \in T_k\ [\forall y \neq x \in T_k \supset \mathrm{tell}(x, y, IR_I)]]$$

Rule 3. After an agent ($x)$ completes an intention (I), it retracts all *IR*s associated with that intention from all team members.

$$\forall x\ [\forall k \ni x \in T_k\ [\forall y \neq x \in T_k \supset \mathrm{tell}(x, y, \neg IR_I)]]$$

Rule 4. Upon generating an intention for a capability (C) an agent (x) tells all team members of any *IR*s associated with that capability.

$$\forall x\ [\forall k \ni x \in T_k\ [\forall y \neq x \in T_k \supset \mathrm{tell}(x, y, IR_C)]]$$

Rule 5. After an agent ($x)$ completes an intention formed for a capability (C), it retracts all *IR*s associated with that capability from all team members.

$$\forall x\ [\forall k \ni x \in T_k\ [\forall y \neq x \in T_k \supset \mathrm{tell}(x, y, \neg IR_C)]]$$

Rule 6. An agent (y) shares all *IR*s it is told about with all teams it is a member of, except the teams from which the originating agent (x) is a member.

$$\forall y[\mathrm{tell}(x, y, IR) \supset \forall k \ni y \in T_k \wedge x \notin T_k\ [\forall z \neq y \in T_k\ [\mathrm{tell}(\underline{y}, z, IR(x,\tau))]]$$

Rule 7. An agent (y) informs all team members of any IR retractions it is told about.

$$\forall y[\mathrm{tell}(x, y, IR) \supset \forall k \ni y \in T_k \wedge x \notin T_k\ [\forall z \neq y \in T_k\ [\mathrm{tell}(\underline{y}, z, \neg IR(x,\tau))]]$$

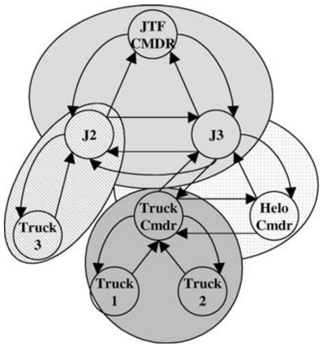
Rule 8. If an agent (x) has a belief ($\bar{\tau}$) that matches the information content of an *IR* from another agent (y), agent x tells agent y belief $\bar{\tau}$, provided agent x has a means for communicating with agent y, otherwise agent x tells the agent (z) which had communicated the *IR*.

$$\begin{aligned}&\forall x[\mathrm{bel}(x,\bar{\tau}) \wedge \mathrm{tell}(z,x,IR(y,\tau)) \wedge \mathrm{match}(\bar{\tau},IR(y,\tau)) \supset \\ &\qquad \text{IF comm}(x,y) \text{ THEN tell}(x,y,\bar{\tau}) \text{ ELSE tell}(x,z,\bar{\tau})]\end{aligned}$$

Figure 6. Truck company commander role definitions (abbreviated).

Truck Agent #3. Truck Agent #3, upon learning any road or bridge state information within the TCC's area of interest will forward that information to the TCC. If, however, the TCC cannot be communicated with directly, the information is forwarded to the J2, who, using the same rules, forwards it to the TCC if possible, otherwise sending it to the J3 who then directs it to the TCC. In this manner, the TCC is not restricted to acquiring route information only from its local team, but may acquire it from agents throughout the organization.

An alternative approach to the problem of disseminating *IR*s and *IP*s is to use middle agents as brokers of these announcements. We plan to explore whether *matchmakers* [23] and/or *facilitators* [24] can be extended to act as collector/providers of *IP*s and *IR*s themselves. In particular, matchmakers are designed to act as yellow pages agents for agent capabilities. It may be very useful to extend the kinds of information about functional capabilities stored in matchmakers to include their associated *IP*s and *IR*s, as a means of



T0) JTFC: $g_1 \rightarrow$ J2; $g_2 \rightarrow$ J3
T1) J2: $\lfloor IP_i^A, IR_i^{J2}, IR_j^{J2} \rfloor \rightarrow$ JTFC, J3, Truck3
T1) J3: $\lfloor IR_i^{J3} \rfloor \rightarrow$ JTFC, J2, TC, HC
T1) TC: $\lfloor IR_j^{TCC}, IP_j^P \rfloor \rightarrow$ J3, HC, Truck1, Truck2
T2) J3: $[S(i, 24, \mathrm{full})] \rightarrow$ J2;
$\lfloor IR_j^{TCC}, IP_j^P \rfloor \rightarrow$ JTFC, J2;
$\lfloor IP_i^A, IR_i^{J2}, IR_j^{J2} \rfloor \rightarrow$ TC, HC
T2) J2: $\lfloor IR_i^{J3} \rfloor \rightarrow$ Truck3
T3) J2: $\lfloor IR_j^{TCC}, IP_j^P \rfloor \rightarrow$ Truck3
T3) TC: $\lfloor IP_i^A, IR_i^{J2}, IR_j^{J2} \rfloor \rightarrow$ Truck1, Truck2

Figure 7. *IP* and *IR* dissemination example.

reducing the amount of traffic resulting from disseminating these announcements. The matchmaker could then inform the potential provider and establish an information channel.

Mixed-Initiative Advising of Extended Activities

Unlike many AI plan representations, shared team plans within organizations often represent large-scale activities with many repetitive elements. The team plans in MIATA, for example, do not specify a sequence of tasks to be accomplished by different agents in series, but rather a set of parallel activities arranged (by support relations) in a workflow network. The team agents handle repetitive subtasks where each cycle supports a subtask of another agent, supporting its overall objective. One of the goals of our model, motivating the use of subscription policies, is to enable agents to reprioritize or reorder their information gathering tasks to support the near-term, higher priority tasks of the agents they are supporting.

As an example, consider what happens when the JTF commander (and his/her PAA) tasks the J3 to focus on delivering supplies to a hard-hit region (Fig. 3). The J3 already has received general direction (by virtue of its role in the team plan) to provide supplies to all who need it in Honduras. The new task directive from the human commander should be interpreted by the PAA as a *reprioritization* of the individual elements of that task, rather than a retasking, so that supplies are provided to the region identified first. When this reprioritized tasking for the J3 is received, it already has subscribed to the information the J2 is collecting about which towns in Honduras need supplies. It now issues a new subscription message for the same information, but with a higher priority, and limits the scope of information needed to the region it must now address first. The result should be a corresponding reprioritization and rescheduling of J2's information gathering activities to support this near-term need.

We are still in the process of formalizing this kind of interaction, as it requires a more complex model of the agents' involved, and the relationships between their tasks. The staff agents must be able to dynamically reorder their pending activities based not only on their own internal priorities, but those of the team members that they are supporting, and the team leader that is tasking them all. We mention it here because we see it as an instance of one of the fundamental issues for mixed-initiative control of agent teams. Humans controlling teams of agents on large-scale tasks need to be able to redirect the agents by reference to collections of subtasks that are shared across team members, as well as by delineating which individual agent on the team is to have a particular responsibility.

Failure Reporting and Failure Handing

Another key long-term issue for the use of agents in mixed-initiative organizations is the effective reporting of failures. As in scheduling and other large constraint systems, failures resulting from the absence of low-level action preconditions can be attributed to many different, indirect causes. A truck, for example, might fail to deliver cargo because a road was blocked, but did that happen because the truck agent did not receive available information about that road, or because the information was not known by the organization? One problem is potentially fixable, while the other must be addressed strictly in terms of searching for a new delivery plan. Furthermore, what is the role of a commander several steps removed from that event? Should he or she be told about every small failure, and if not, when should agents report such failures up to the next level in the organization? Suppose ten trucks fail because of the same blocked road. This would be evidence of an information flow failure. But if the truck company commander came to know that most of the roads in a particular region were blocked, or all roads leading to a town were blocked, this would indicate that the region had suffered severe damage, and helicopters (from another team) should be used for the deliveries.

In both cases, we have repeated domain failures that seem to require promotion of the failure to a higher level within the organization. Both may have a similar domain-level solution for the original objective at that level (deliver the supplies to towns), such as by using helicopters. However, one may also indicate an agent team communications failure. How can we design the agents to effectively cooperate with the humans operating at various levels in the organization to begin to address these problems? While there is no single answer to this question, we are beginning to experiment with several strategies. First, the software agents must have the ability to recognize their failures and report them, together with some indication of the cause (such as a missing precondition). The information flow policies described in this paper support some appropriate reporting mechanisms for this. Each agent capability is associated with a default policy as to who should receive task failure messages, as well as in-progress status messages. Our current approach is to have failure messages be reported in full up to the nearest agent supported by a human. Status messages reported by team leader PAA's to their human controllers need to be more tightly controlled so as not to swamp the user. However, periodic status reports to the user can include *summaries* (rollups) of the intermediate failures of subordinate agents, and provide dialog support for drilling down on those summaries when the human commander requests it. Graphical analysis aids are important here. For example, showing the locations of blocked roads on a map (similar to Fig. 3(B)) might lead a human commander to quickly reassign all deliveries in a region to be handled by the helicopter company.

Second, we must find ways for agents to handle as many failures as possible by retasking or replanning within local teams. We are currently using several scheduling support agents in the MIATA system with this capability. The DEO scheduler from CIRL, for example, can rebuild schedules quickly for tasks affected by blocked roads or delays, when new information about those events is provided. In this case, a summary, and possibly some analysis, of the tasks that could not be handled is what should be reported to the commander. Other reactive execution agents may try multiple approaches to achieving an objective before failing entirely. However, with all of these ways of making agents more robust, we need to be aware of the conditions under which they should be reporting their initial failures even if they ultimately succeed.

Finally, we need to address the question of how PAAs can help users in replanning, given evidence of failures of different kinds. As subordinate agents fail on tasks or exceed the expected time to complete, managers may have a choice whether to give more time, find additional resources, retask to other agents, perform tasks themselves, etc. We need to find ways to assist users in chosing how to proceed. The approach taken by Klein and Dellarocas [25, 26] seems promising here at several levels. They are compiling a taxonomy of systemic agent failures and related kinds of workflow systems failures, each linked to possible repair approaches. Whether or not the corresponding repairs can be accomplished automatically or not, it will be helpful to a user to know what some possible diagnoses of the problems are, and PAA's may be able to assist in implementing repairs from such a library, with advice from the user. Finally, while many of the problems may be traceable to workflow interaction failures, many others will be domain specific. For this, domain models will be required to help in the diagnosis of both execution failures and planning failures [27].

Conclusions

We have presented a framework for dynamic information sharing among teams of humans and agents working as parts of larger, frequently hierarchical, organizations. The model was motivated by our work to develop approaches to mixed-initiative human/computer control of agent-based systems, and in particular from our role in the development, in coordination with a large team of other researchers, of the MIATA demonstration of mixed-initiative agent teams in a hurricane disaster relief scenario. Our key observation is that information needs, as well as agent intentions to execute plans and achieve goals. In many cases, the information sharing of information needs and provision capabilities requires wider dissemination than the sharing of team intentions. 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 mixed-initiative command and control of such systems as it removes the burden from users who might otherwise need to explicitly characterize which agents needed to communicate with other agents.

Acknowledgments

This work was funded under DARPA contract F30602-98-C-0168. We thank Prof. James Hendler, DARPA Program Manager for CoABS for organizing the MIATA group and giving us tremendous latitude in developing the project.

Notes

1. The MIATA working group consisted of Mark Burstein, David Diller, Alice Mulvehill, Brett Benyo and Ed Pattison-Gordon (BBN Technologies), Karen Myers and David Moreley (SRI), George Ferguson and James Allen (University of Rochester), Sebastian Thrun and Jaime Schulte (CMU), Brian Drabble and Najam-ul Haq (CIRL, University of Oregon), Drew McDermott (Yale University), Doug Smith and Stephen Westfold (Kestrel Institute), and Steve Ford (OBJS, Inc.). We are indebted to everyone for their hard work on this project.
2. The CAMPS Mission Planner [11] is a scheduling system for cargo aircraft being developed by BBN and Kestrel Institute for the Air Force. The JADE logistics planner [12], developed at MITRE and BBN, is a prototype interactive case-based reasoning tool for composing deployment plans consisting of lists of cargo to be picked up from depots and delivered to the theater. Both were initially stand-alone systems with graphical user interfaces that were adapted for this work. CIRL's DEO scheduler was wrapped as an agent without a user interface to provide the functionality required to schedule the deliveries and other activities of trucks and helicopters.

References

1. D.A. Norman, "How people might interact with agents," in *Software Agents*, edited by J. Bradshaw, MIT Press: Cambridge, MA, 1997, pp. 49–55.
2. M.H. Burstein, G. Ferguson, and J. Allen, "Integrating agent-based mixed-initiative control with an existing multi-agent planning system," in *Proc. Int. Conf. On MultiAgent Systems*, 2000.
3. M. Burstein and D. McDermott, "Issues in the development of human-computer mixed-initiative planning systems," in *Cognitive Technology*, edited by B. Gorayska and J.L. Mey, Elsever Science, 1996, pp. 285–303.
4. M.H. Burstein, A. M. Mulvehill, and S. Deutsch, "An approach to mixed-initiative management of heterogeneous software agent teams," in *Proc. 32nd Hawaii Int. Conf. on System Sciences*, 1998.
5. T. Winograd and F. Flores, *Understanding Computers and Cognition: A New Foundation for Design*, Addison-Wesley: Reading, MA, 1987.
6. K.L. Myers and P.M. Berry, "At the boundary of workflow and AI," in *Proc. AAAI 1999 Workshop on Agent-Based Systems in The Business Context*, 1999.
7. R.J. Firby, "Task networks for controlling continuous processes," in *Proc. Int. Conf. AI Plannng Systems*, Meno Park, CA, 1994.
8. M. Tambe, "Towards flexible teamwork," *J. Artif. Intell. Research*, vol. 7, pp. 83–124, 1997.
9. B.J. Grosz and S. Kraus, "Collaborative plans for complex group action," *Artif. Intell.*, vol. 86, pp. 269–357, 1996.
10. B.J. Grosz and S. Kraus, "The evolution of SharedPlans," in *Foundations and Theories of Rational Agency*, edited by M. Wooldridge and A. Rao, 1999, pp. 227–262.
11. M. Burstein and T. Emerson, "Development of a constraint-based airlift scheduler by program synthesis from formal specifications," in *Proceedings of the 1999 Conf. on Automated Software Engineering*, Orlando, FL, 1999.
12. A. Mulvehill and J. Caroli, "JADE: A tool for rapid crisis action planning," in *Proc. of the 1999 Command and Control Research and Technology Symposium*, US Naval War College, Newport, RI, 1999.
13. S.E. Deutsch, "Interdisciplinary foundations for multiple-task human performance modeling in OMAR," in *Proc. Cognitive Science Society*, Madison, WI, 1998.
14. K.L. Myers, "User's guide for the procedural reasoning system," *Artif. Intell. Center*, SRI International: Menlo Park, CA, 1993.
15. GITI/ISX Grid Vision Team, "The CoABS Grid: Technical Vision," http://coabs.gobalinfotek.com, 2000.
16. K. Arnold, B. O'Sullivan, R.W. Schiefler, J. Waldo, and A. Wollrath, *The Jini Specification. The Jini Technology Series*, Addison-Wesley: Reading, MA, 1999.
17. M. Tambe and W. Zhang, "Towards flexible teamwork in persistent teams: Extended report," *Autonomous Agents and Multi-agent Systems*, vol. 3, pp. 159–183, 2000.
18. P.R. Cohen and H.J. Levesque, "Teamwork," Nous 25(4), Special Issue on *Cognitive Science and Artif. Intell.*, pp. 487–512, 1991.
19. P.R. Cohen, H.R. Levesque, and I. Smith, "On team formation," in *Contemporary Action Theory*, edited by J. Hintikka and R. Tuomela, Synthese, 1997
20. J.M. Bradshaw, S. Dutfield, P. Benoit, and J.D. Woolley, "KAoS: Toward an industrial-strength open agent architecture," in *Software Agents*, edited by J. Bradshaw, Cambridge, MA: MIT Press, 1997, pp. 375–418.
21. G. Ferguson and J. Allen, "TRIPS: An integrated intelligent problem-solving assistant," in *Proc. National Conf. Artif. Intell.*, Madison, WI, 1998, pp. 567–573.
22. K.L. Myers, "Advisable planning systems," in *Advanced Planning Technology*, edited by A. Tate, AAAI Press: Menlo Park, CA 1996.
23. K. Decker, K. Sycara, A. Pannu, and M. Williamson, "Designing behaviors for information agents," in *Proc. of the First International Conference on Autonomous Agents*, Feb. 1997.

24. A. Cheyer and D. Martin, "The open agent architecture" *JAMS*, vol. 4, no. 1–2, pp. 143–148, 2001.
25. M. Klein and C. Dellarocas, "Exception handling in agent systems," in *Proc. Third Int. Conf. on Autonomous Agents*, Seattle, WA, May, 1999.
26. M. Klein and C. Dellarocas, "A knowledge-based approach to handling exceptions in workflow systems," *J. Computer-Supported Collaborative Work*. Special Issue on *Adaptive Workflow Systems*, vol. 9, 2000.
27. L. Birnbaum, G. Collins, M. Freed, and B. Krulwich, "Model-based diagnosis of planning failures," in *Proc. Eighth National Conf. on Artif. Intell.*, 1990.

Mark H. Burstein is a Division Scientist and the Director of the Human Centered Systems Group at BBN Technologies. He has been at BBN since 1984. He holds a B.S. in Mathematics from M.I.T., and a Ph.D. in Computer Science (Artificial Intelligence) from Yale University. He is the author or co-author of over 50 papers in a number of areas, including memory-based reasoning for natural language processing, modeling of human plausible and analogical reasoning, knowledge representation and knowledge acquisition, machine learning, mixed-initiative planning and scheduling, multi-agent systems, and semantic web services.

David E. Diller is a Senior Scientist at BBN Technologies in Cambridge, MA. His current focus includes cognitive modeling, mixed-initiative agent based systems, and simulation-based training applications. Dr. Diller holds a Ph.D. in Cognitive Science and Psychology and a M.S. in Computer Science from Indiana University.