Coordination in Human-Agent-Robot Teamwork

Jeffrey M. Bradshaw, Paul J. Feltovich, Matthew Johnson, Larry Bunch, Maggie Breedy, Hyuckchul Jung, James Lott, and Andrzej Uszok

Florida Institute for Human and Machine Cognition (IHMC)
40 South Alcaniz Street, Pensacola, FL 32502 USA +1 (850) 202-4462
{jbradshaw, pfeltovich, mjohnson, lbunch, mbreedy, hjung, jlott, auszok}@ihmc.us

Abstract

Coordination is an essential ingredient of a *teamwork-centered* approach to autonomy. In this paper, we discuss some of the challenges and requirements for successful coordination, and briefly how we have used KAoS HART services to support coordination in a multi-team human-robot field exercise.

Teamwork-Centered Autonomy

Planning technologies for intelligent systems often take an *autonomy-centered* approach, with representations, mechanisms, and algorithms that have been designed to accept a set of goals, and to generate and execute a complete plan in the most efficient and sound fashion possible. While this approach may be the best choice for situations where it is impractical or impossible for humans to provide close supervision of the intelligent system (e.g., [28]), it is not sufficient for the increasing number of applications that require close and continuous interaction with people and with other autonomous components (e.g., [10; 29]).

A *teamwork-centered autonomy* approach takes as a beginning premise that people are working in parallel alongside one or more autonomous systems, and hence adopts the stance that the processes of understanding, problem solving, and task execution are necessarily incremental, subject to negotiation, and forever tentative. Thus, a successful approach to teamwork-centered autonomy will require that autonomous systems be designed to facilitate the kind give-and-take and richness of interaction that characterize natural and effective teamwork among groups of people [4].[1]

[1] There are important differences between human teams and the mixed teams of which we write, leading some to wonder whether the use of the term "team" is appropriate in this context and whether machines and software can appropriately be classed as "team members." While recognizing the significant—and perhaps insurmountable—differences between the contributions that technology and people can make to joint activity, a large segment of the research community has concluded that using "team" is appropriate as a rough way of characterizing the ideal forms of interaction to which we aspire. For recent snapshots of this ongoing debate, see [2; 3; 13].

Over the past several years, we have been interested in learning how to facilitate such teamwork among humans, agents, and robots. To lay the groundwork for our research, we have studied how humans succeed and fail in joint activity requiring a high degree of interdependence among the participants [14; 24]. Such interdependence requires that, in addition to what team members do to accomplish the work itself, they also invest time and attention in making sure that distributed or sequenced tasks are appropriately *coordinated*.

Although there are several important challenges in making automation a team player [25], in this paper we focus on only on the problem of coordination. Following a brief description of this aspect of joint activity, we describe the KAoS HART (Human-Agent-Robot Teamwork) services framework, which has been developed as a means of exploring our ideas about the role of regulatory constraints in joint activity [5; 7; 19; 21; 32]. We give simple examples of some of the kinds of policies we have been exploring. Finally, we discuss a field exercise that allowed us to implement and explore many of these issues. This exercise involved mixed human-robot teams whose objective was to apprehend an intruder hiding on a cluttered Navy pier [23].

Understanding Coordination

The Challenge of Human-Agent Coordination

Malone and Crowston [27] defined coordination as "managing dependencies between activities." Teamwork, which by definition implies interdependence among the players, therefore requires some level of work for each party over and beyond the carrying out of task itself in order to manage its role in coordination. Part of that "extra" work involves each party doing its part to assure that relevant aspects of the agents and the situation are observable at an appropriate level of abstraction and using an effective style of interaction [9].

Although coordination is as much a requirement for agent-agent teamwork as it is for human-agent teamwork, the magnitude of the representational and reasoning gulfs separating humans from agents is much larger. Moreover, because the agent's ability to sense or infer information

about the human environment and cognitive context is so limited, agent designers must find innovative ways to compensate for the fact that their agents are not situated in the human world. Brittleness of agent capabilities is difficult to avoid because only certain aspects of the human environment and cognitive context can be represented in the agent, and the representation that is made cannot be "general purpose" but must be optimized for the particular use scenarios the designer originally envisioned. Without sufficient basis for shared situation awareness and mutual feedback, coordination among team members simply cannot take place, and, of course, this need for shared understanding and feedback increases as the size of the team and the degree of autonomy increase.

Notwithstanding these challenges, adult humans and radically less-abled entities (e.g., small children, dogs, video game characters) are capable of working together effectively in a variety of situations where a subjective experience of collaborative teaming is often maintained despite the magnitude of their differences. Generally this is due to the ability of humans to rapidly size up and adapt to the limitations of their teammates in relatively short order, an ability we would like to exploit in the design of approaches for human-agent teamwork.

Requirements for Effective Coordination

There are three requirements for effective coordination: interpredictability, common ground, and directability [24]:

- *Interpredictability:* In highly interdependent activities, it becomes possible to plan one's own actions (including coordination actions) only when what others will do can be accurately predicted. Skilled teams become interpredictable through shared knowledge and idiosyncratic coordination devices developed through extended experience in working together; bureaucracies with high turnover compensate for experience by substituting explicit predesigned structured procedures and expectations.
- *Common ground:* Common ground refers to the pertinent mutual knowledge, beliefs, and assumptions that support interdependent actions in the context of a given joint activity [15]. This includes initial common ground prior to engaging in the joint activity as well as mutual knowledge of shared history and current state that is obtained while the activity is underway. Unless I can make good assumptions about what you know and what you can do, we cannot effectively coordinate.
- *Directability:* Directability refers to the capacity for deliberately assessing and modifying the actions of the other parties in a joint activity as conditions and priorities change [12]. Effective coordination requires responsiveness of each participant to the influence of the others as the activity unfolds.

Following the lead of pioneering researchers such as Geertz [20, pp. 44-46, 67], we have argued that people create and have created cultures and social conventions—albeit in many disparate forms across mankind that can be hard for outsiders to understand—to provide order and predictability that lead to effective coordination [17; 19], including ongoing progress appraisal [18].[1] Order and predictability may have a basis in the simple cooperative act between two people, in which the parties "contract" to engage together in a set of interlinked, mutually beneficial activities. From this simple base, in humans at least, there are constructed elaborate and intricate systems of regulatory tools, from formal legal systems, to standards of professional practice, to norms of proper everyday behavior (along with associated methods of punishment or even simple forms of shaming for violations of these).

People coordinate through signals and more complex messages of many sorts (e.g., face-to-face language, expressions, posture). Human signals are also mediated in many ways—for example, through third parties or through machines such as telephones or computers. Hence, direct and indirect party-to-party communication is one form of a "coordination device," in this instance coordination by *agreement*. For example, a group of scientists working together on a grant proposal, may simply agree, through e-mail exchanges, to set up a subsequent conference call at a specific date and time. There are three other major types of coordination devices that people commonly employ: *convention*, *precedent*, and *situational salience* [14; 24].

Roles can be thought of as ways of packaging rights and obligations that go along with the necessary parts that people play in joint activities. Knowing one's own role and the roles of others in a joint activity establishes expectations about how others are likely to interact with us, and how we think we should interact with them. Shoppers expect cashiers to do certain things for them (e.g., total up the items and handle payment) and to treat them in a certain way (e.g., with cheerful courtesy), and cashiers have certain expectations of shoppers. When roles are well understood and regulatory devices are performing their proper function, observers are likely to describe the activity as highly-coordinated. On the other hand, violations of the expectations associated with roles and regulatory structures can result in confusion, frustration, anger, and a breakdown in coordination.

Collections of roles are often grouped to form organizations. In addition to regulatory considerations at the level of individual roles, organizations themselves may also add their own rules, standards, traditions, and so forth, in order to establish a common culture that will smooth interaction among parties.

Knowing how roles undergird organizations and how rights and obligations undergird roles helps us understand how organizations can be seen as functional or dysfunctional. Whether hierarchical or heterarchical, fluid or relatively static, organizations are functional only to the extent that their associated regulatory devices and roles

[1] Even simple forms of animal cooperation seems to bear out such a thesis [30], and we would argue that the more autonomous the agents involved, the more need there is for such regulation and the wider the variety of forms it might take.

generally assist them in facilitating their individual responsibilities and their work in coordinating their actions with others when necessary.

The lesson here for mixed human-agent-robot teams is that the various roles that team members assume in their work must include more than simple names for the role and algorithmic behavior to perform their individual tasks. They must also, to be successful, include regulatory structures that define the additional work of coordination associated with that role.

The KAoS HART Services Framework

The KAoS HART (Human-Agent-Robot Teamwork) services framework has been adapted to provide the means for dynamic regulation on a variety of agent, robotic, Web services, Grid services, and traditional distributed computing platforms [5; 21-23; 26; 29; 32]. It also provides the basic services for distributed computing, including message transport and directory services, as well as more advanced features like domain and policy services.

In addition to the considerations mentioned above, our research has been guided by three principles. First, we focus on situations where it is desirable for humans to remain "in-the-loop" and allow the degree and kind of control exercised by the human to vary at the initiative of the human or, optionally, with automated assistance [6; 8]. Second, we assure that mechanisms for appropriate robot regulation, communication, and feedback in such situations are included from the start in the foundations of system design, rather than layered on top as an afterthought [21]. Third, working in the tradition of previous agent teamwork researchers (e.g., [16] [31]), we attempt to implement a reusable model of teamwork involving a notion of shared knowledge, goals, and regulatory mechanisms that function as the glue that binds team members together. The KAoS services framework, which implements this reusable model, is described in the next section.

All team members, human and agent, register with the directory service and provide a description of their capabilities. This enables team members to query the directory service to find specific team members as well as match them based on capability. The domain and policy services manage the organizational structure among the agents, providing the specification of roles and allowing dynamic team formation and modification. A"KAoS Robot" extension [21] provides a generic wrapper for each type of robot and a consistent interface for client systems to access the robots. KAoS Robot enables detailed status monitoring in addition to policy checking and enforcement, providing essential ingredients for coordination.

Policies, implementing coordination constraints, are implemented in OWL (Web Ontology Language: http://www.w3.org/ 2004/OWL), to which we have added optional extensions to increase expressiveness (e.g., role-value maps) [32]. A growing set of services for policy deconfliction and analysis are also provided [11; 32]. Policies are used to dynamically regulate the behavior of system components without changing code or requiring the cooperation of the components being governed. By changing policies, a system can be continuously adjusted to accommodate variations in externally imposed constraints and environmental conditions. There are two main types of polices; authorizations and obligations. The set of permitted actions is determined by *authorization policies* that specify which actions an actor or set of actors are permitted (*positive authorizations*) or not allowed (*negative authorizations*) to perform in a given context. *Obligation policies* specify actions that an actor or set of actors is required to perform (*positive obligations*) or for which such a requirement is waived (*negative obligations*). From these primitive policy types, we build more complex structures that form the basis for team coordination.

Coordination Policy Examples

Cohen-Levesque Notification Obligation Policy

One of the most well known heuristics in team coordination was originally formulated by Cohen and Levesque as follows: "any team member who discovers privately that a goal is impossible (has been achieved, or is irrelevant) should be left with a goal to make this fact known to the team as a whole" [16, p. 9]. For example, if a robot were asked to get a wrench from the garage so it can help fix a sink, and the robot finds the garage door locked, it would be expected to inform the other partner about this. We have implemented our version of this heuristic in the form of an obligation policy that can be roughly described as follows:

A Robot is obligated to notify it Teammates (in this case, the Requestor is the only teammate) when Action is Finished (whether Successfully Completed, Aborted, or Irrecoverably Failed)

When the robot encountered the locked door, its navigation task would fail and trigger the obligation.

Runtime Policy Addition and Modification

KAoS provides a mechanism to support runtime addition and modification of policies in support of coordination. For example, for a joint tracking task one partner may want to know when the other partner has acquired the target so he or she can disengage and reposition. Using the TRIPS dialogue capability [1] integrated with KAoS, the operator might simply state "Let me know when you see the target" in order to establish a one-time obligation. We have created a standing obligation policy for our robots that triggers a message stating "I see the target" when the target detection module determines that the target has been identified. Such a policy could be established by saying, "Always let me know when you see a target for the first time."

Acknowledgements and Policy Deconfliction

We implemented a basic policy that requires robots to acknowledge requests. While this seemed a good general rule, there are important exceptions that need to be handled through KAoS policy deconfliction capabilities [11].

One reasonable exception to the acknowledgement policy is that people do not always verbally acknowledge requests, particularly when they are directly observable. Direct observability means that when a human requestor sends the communication to a robot receiver, the fact that the request was received, understood and being acted upon is observable by the requestor. For example, when a robot is told to move forward five meters, and then can be seen starting to move forward, there is normally no need for the robot to state "I have received your request to move forward and have begun." The same applies to queries. When somebody asks a robot "where are you," it is unnecessary for it to reply "I have heard your question and am about to reply", if it alternatively simply says "in the library." We implemented two additional policies to waive the obligation to acknowledge requests when the request is either a teleoperation command or a query.

<u>*Acknowledgement Policy Set*</u>

1) A Robot is obligated to acknowledge to the Requestor when the Robot Accepts an Action

2) A Robot is not obligated to acknowledge Teleoperation requests

3) A Robot is not obligated to acknowledge Query requests

The two policies do indeed conflict with the original, but by assigning the more restrictive polices a higher priority (which can be done numerically or logically), it is possible to automatically deconflict these policies and achieve the desired behavior.

Role Management and Progress Appraisal

Groups often use roles to perform task division and allocation. Roles provide a membership-based construct with which to associate sets of privileges (authorizations) and expected behaviors (obligations). When an actor is assigned to a role, the regulations associated with the role automatically apply to the actor and, likewise, are no longer applicable when the actor relinquishes the role. These privileges and expectations that comprise a role may be highly domain dependent. For example the role "Team Leader" in a military domain is significantly different from "Team Leader" in sports. Roles may also specify expected behaviors. For example, if your role is a "Sentry" then you are obligated to remain at your post, and other actors will expect you to fulfill that obligation. Roles can also affect other behaviors such as expected communications. If you are assigned to be a "Sentry", you are obligated to announce any violations of your boundary and report these to your immediate superior.

Taking advantage of the extensibility and inheritance properties of OWL ontologies, roles can be defined at various levels of abstraction with sub-roles refining the regulations pertinent to more generic super-roles. In this way, some high-level roles need not be domain specific or involve specific tasking, but they are still defined by their associated regulations. "Teammate" can be considered a generic role that has some of its regulations already noted. We view this level of abstraction as appropriate for expectations that facilitate coordination such as acknowledgements and progress appraisals. The obligation to acknowledge requests can be thought of as a policy associated with being a teammate. We have developed two policy sets that we feel apply generally to robots assigned to the role of "Teammate." The first is the acknowledgement policy set discussed above. The second involves progress appraisal:

<u>*Progress Appraisal Policy Set*</u>

1) A Robot is obligated to notify the Requestor when requested Action is Finished (includes Completed, Aborted, and Failure).

2) A Robot is not obligated to notify the Requestor when a requested Tele-operation Action is Completed.

3) A Robot is not obligated to notify the Requestor when a requested Query Action is Completed.

The first policy ensures that the requestor of a task is notified when the tasked robot encounters problems or successfully completes the task since the action status of Finished is ontologically defined as a super-class of the statuses Completed, Failed, and Aborted. The second two policies in this set are exceptions similar to those in the acknowledgement set. With knowledge that these policies are in place, human and robotic team members have the mutual expectation that these progress appraisals will be performed. There is no longer a need to explicitly ask for such communication and, perhaps just as importantly, the absence of these obligatory communications becomes an indicator that additional coordination may be necessary. For example, I command a robot to autonomously navigate to a distant location. Since I know the robot would notify me if it had arrived or it was stuck or had otherwise failed, I can assume that it is still moving toward the goal. If I was concerned with an approaching deadline or that the task was taking too long, I would query for the robot's position and create a new estimate of when it should reach the goal.

The policies outlined here are just one of several sets that we have explored, informed on previous theoretical work and simulations and field experiments performed by ourselves and by others [1, 3, 15-17, 19, 26-28]. As we encounter new challenges in future work, we will continue to revise and expand such policy sets.

Policies Relating to Team Leaders

In contrast to our previous work on human-robot teams where all team members were "equal," we decided to explore the idea of team "leaders." Leaders not only must adhere to their own regulations, but they also impact the regulatory structure of all the other roles in the group. Peer

interaction may be undirected, but Leaders tend to alter the pattern of activity, with themselves becoming the focal point. In particular we have identified several policy sets particular to leaders. The first set is about the chain of command:

Chain of Command Policy Set
1) A Robot is authorized to perform Actions requested by its Team Leader
2) A Robot is authorized to Accept Actions requested by a higher authority
3) A Robot is not authorized to perform Action requests from just any Requestor
4) A Robot is authorized to Accept Actions that are self-initiated

The first policy gives team leaders the authority to command their team. The second gives the same authority to anyone directly higher in the chain of command. The third policy explicitly restricts access to the robots from outside of the chain of command. The fourth policy makes self initiated actions an exception to the third policy.

Another set was used to explore notification to help maintain common ground among team members:

Notification Policy Set
1) A Robot is obligated to notify its Team Leader when an Action is requested by a higher authority
2) A Robot is obligated to notify Its Team Leader when starting a self-initiated Action
3) A Robot is obligated to notify its Team Leader when a self-initiated Action is Finished (includes statuses of Completed, Aborted, and Failure).

Team Creation and Management

The KAoS Directory Service manages organizational structure, allowing dynamic team formation and modification. Teams and subteams can be created dynamically, allowing for the creation of complex organizational structures. Agents can join and leave teams as necessary to support the desired structure. Actors can be assigned roles including Team Leader, affecting the dynamics of coordination as discussed in the previous section. Queries can be made to identify current team structure, who is on a certain team currently, or who is team leader. In the next section we describe a demonstration that highlights the creation of organizational structure, in this case a hierarchical team such as found in the military. It also embodies dynamic team composition and fluid assignment of roles.

The Coordinated Operations Exercise

Mission Scenario

Consider a scenario in which an intruder must be discovered and apprehended on a cluttered Navy pier (figure 1). To support the search, you can draw on the abilities of an additional human and five robots. While there are plenty of issues to address including robot capabilities, sensor limitations, and localization, we focused on the coordination aspects of the task. We specifically designed the task to have more robots than a single individual could easily handle by teleoperation. We also wanted to make sure the scenario included more than one human, since this provides its own challenges.

Figure 1. The pier

Team Composition

The available team members consisted of two humans and five robots (figure 2). The humans were to play distinct roles. One was the "Commander" who was to establishing subteams and manage the overall search process. Relying on a combined speech and graphical interface the Commander operated remotely without direct sight of the area of operation. The second human played the role of "Lieutenant." The Lieutenant would be assigned to a team just like the robots and he worked in the field generally alongside and in sight of them. He wore a backpack that carried a laptop to provide a similar speech and visual interface as the Commander's, through a head mounted display as shown in figure 2. The robot team members included four Pioneer 3AT robots variously equipped with different combinations of sonar, GPS, pan-tilt-zoom cameras, and SICK lasers. The fifth robot was an IHMC-designed and -built robot called the tBot. All the robots had onboard computers and used wireless routers for communication.

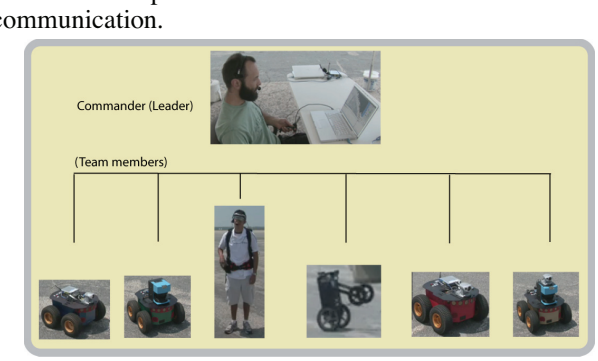

Figure 2. Initial two-tier hierarchical team structure

All of the previously discussed policy sets, including acknowledgement, progress appraisal, notification, and chain of command, were in force for the exercise.

Mission Execution

The Commander must first secure the area boundaries, and forms two subteams to block the two possible avenues of escape. Using natural language, the Commander composed two teams and assigned leaders for each of them (figure 3). One team (Team Alpha) was fully robotic, two robots with one assigned as the leader. The other team (Team Bravo) was mixed, two robots with the Lieutenant assigned to lead. Acknowledgement policies provided useful feedback to the Commander that teams had been successfully formed, since there was no external indication of the fact.

The Commander next defined an area of interest on his display and tasked each team to secure a particular side. After issuing the commands, the Commander dynamically created an obligation policy through speech to be notified by the team leaders when each team was in position. Once in position, the coordination policy took effect and the robot team leader reported.

The boundaries having been made secure, the Commander directed each team to begin a search of the area. The autonomous team began to search under the direction of its robotic team leader. The Lieutenant used natural language to direct his team for the search. When the intruder was found by a robot, the appropriate team leader was informed according to existing coordination policies.

To apprehend the intruder, the Lieutenant tried to use the tBot, a robot not currently assigned to his team (figure 3). The coordination services enforced the chain of command and prevented the action. The Lieutenant then proceeded through the policy-required chain of command to acquire permission—i.e., he asked the Commander. The Commander dynamically assigned the tBot to the Lieutenant's team. The Lieutenant was now authorized to make use of the tBot, and the apprehension was successful. Notice that the dynamic assignment of an agent to a certain group automatically brought with it all of that group's extant regulatory structure, including the authority for that group's leader to give orders to his charges.

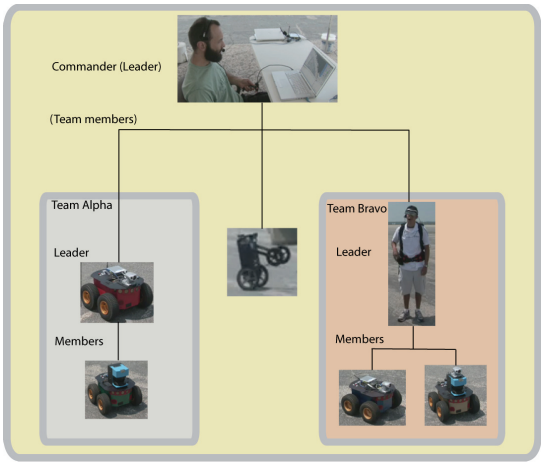

Figure 3. Three tier hierarchical team composed of two subteams (tBot still on original team).

References

1. Allen, J.F., D.K. Byron, M. Dzikovska, G. Ferguson, L. Galescu, and A. Stent. "Towards conversational human-computer interaction." *AI Magazine 22*, no. 4 (2001): 27-35.
2. Biever, C. "'Bots as peers: It's all a matter of teamwork." San Francisco, CA: San Francisco Chronicle, May 6, 2007. http://sfgate.com/cgi-bin/article.cgi?f=/c/a/2007/05/06/ING9GPK9U71.DTL. (accessed September 15, 2007).
3. Biever, C. "If you're happy, the robot knows it." *The New Scientist 193*, no. 2596 (March 24 2007): 30-31. http://technology.newscientist.com/article/mg19325966.500-if-youre-happy-the-robot-knows-it.html. (accessed September 15, 2007).
4. Bradshaw, J.M., A. Acquisti, J. Allen, M.R. Breedy, L. Bunch, N. Chambers, P. Feltovich, L. Galescu, M.A. Goodrich, R. Jeffers, M. Johnson, H. Jung, J. Lott, D.R. Olsen Jr., M. Sierhuis, N. Suri, W. Taysom, G. Tonti, and A. Uszok. "Teamwork-centered autonomy for extended human-agent interaction in space applications." Presented at the AAAI 2004 Spring Symposium, Stanford University, CA, 22-24 March, 2004.
5. Bradshaw, J.M., P. Beautement, M.R. Breedy, L. Bunch, S.V. Drakunov, P.J. Feltovich, R.R. Hoffman, R. Jeffers, M. Johnson, S. Kulkarni, J. Lott, A. Raj, N. Suri, and A. Uszok. "Making agents acceptable to people." In *Intelligent Technologies for Information Analysis: Advances in Agents, Data Mining, and Statistical Learning*, edited by N. Zhong and J. Liu, 361-400. Berlin: Springer Verlag, 2004.
6. Bradshaw, J.M., P. Feltovich, H. Jung, S. Kulkarni, W. Taysom, and A. Uszok. "Dimensions of adjustable autonomy and mixed-initiative interaction." In *Agents and Computational Autonomy: Potential, Risks, and Solutions. Lecture Notes in Computer Science, Vol. 2969*, edited by M. Nickles, M. Rovatsos, and G. Weiss, 17-39. Berlin, Germany: Springer-Verlag, 2004.
7. Bradshaw, J.M., P.J. Feltovich, H. Jung, S. Kulkarni, J. Allen, L. Bunch, N. Chambers, L. Galescu, R. Jeffers, M. Johnson, M. Sierhuis, W. Taysom, A. Uszok, and R. Van Hoof. "Policy-based coordination in joint human-agent activity." Presented at the Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, The Hague, Netherlands, 10-13 October, 2004.
8. Bradshaw, J.M., H. Jung, S. Kulkarni, M. Johnson, P. Feltovich, J. Allen, L. Bunch, N. Chambers, L. Galescu, R. Jeffers, N. Suri, W. Taysom, and A. Uszok. "Toward trustworthy adjustable autonomy in KAoS." In *Trusting Agents for Trustworthy Electronic Societies*, edited by R. Falcone. Berlin: Springer, 2005.
9. Bradshaw, J.M., M. Sierhuis, A. Acquisti, P. Feltovich, R. Hoffman, R. Jeffers, D. Prescott, N. Suri, A. Uszok, and R. Van Hoof. "Adjustable autonomy and human-agent teamwork in practice: An interim report on space applications." In *Agent*

Autonomy, edited by H. Hexmoor, R. Falcone, and C. Castelfranchi, 243-280. Kluwer, 2003.

10. Bradshaw, J.M., M. Sierhuis, Y. Gawdiak, R. Jeffers, N. Suri, and M. Greaves. "Teamwork and adjustable autonomy for the Personal Satellite Assistant." Presented at the Workshop on Autonomy, Delegation and Control: Interacting with Autonomous Agents. Seventeenth International Joint Conference on Artivicial Inelligence (IJCAI-2001), Seattle, WA, 6 August, 2001.
11. Bradshaw, J.M., A. Uszok, R. Jeffers, N. Suri, P. Hayes, M.H. Burstein, A. Acquisti, B. Benyo, M.R. Breedy, M. Carvalho, D. Diller, M. Johnson, S. Kulkarni, J. Lott, M. Sierhuis, and R. Van Hoof. "Representation and reasoning for DAML-based policy and domain services in KAoS and Nomads." Presented at the Proceedings of the Autonomous Agents and Multi-Agent Systems Conference (AAMAS 2003), Melbourne, Australia, 14-18 July, 2003, 835-842.
12. Christofferson, K. and D.D. Woods. "How to make automated systems team players." In *Advances in Human Performance and Cognitive Engineering Research, Vol. 2*, edited by E. Salas. JAI Press, Elsevier, 2002.
13. Clancey, W.J. "Roles for agent assistants in field science: Understanding personal projects and collaboration." *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews 32*, no. 2 (2004).
14. Clark, H.H. *Using Language*. Cambridge, UK: Cambridge University Press, 1996.
15. Clark, H.H. and S.E. Brennan. "Grounding in communication." In *Perspectives on Socially Shared Cognition*, edited by L.B. Resnick, J.M. Levine, and S.D. Teasley. Washington, D.C.: American Psychological Association, 1991.
16. Cohen, P.R. and H.J. Levesque. "Teamwork." Menlo Park, CA: SRI International, 1991.
17. Feltovich, P., J.M. Bradshaw, W.J. Clancey, and M. Johnson. "Toward and Ontology of Regulation: Support for Coordination in Human and Machine Joint Activity." In *Engineering Societies for the Agents World VII*, edited by G. O'Hare, et al., in press. Heidelberg, Germany: Springer-Verlag, 2007.
18. Feltovich, P., J.M. Bradshaw, W.J. Clancey, M. Johnson, and L. Bunch. "Progress appraisal as a challenging element of coordination in human and machine joint activity." Presented at the Engineering Societies for the Agents World VIII, Athens Greece, October, 2007.
19. Feltovich, P., J.M. Bradshaw, R. Jeffers, N. Suri, and A. Uszok. "Social order and adaptability in animal and human cultures as an analogue for agent communities: Toward a policy-based approach." In *Engineering Societies in the Agents World IV. LNAI 3071*. Lecture Notes in Computer Science, 21-48. Berlin, Germany: Springer-Verlag, 2004.
20. Geertz, C. *The Interpretation of Cultures*. New York, NY: Basic Books, 1973.
21. Johnson, M., J.M. Bradshaw, P. Feltovich, R. Jeffers, and A. Uszok. "A semantically-rich policy-based approach to robot control." In *Proceedings of the Intenational Conference on Informatics in Control, Automation, and Robotics*. Lisbon, Portugal, 2006.
22. Johnson, M., P. Chang, R. Jeffers, J.M. Bradshaw, V.-W. Soo, M.R. Breedy, L. Bunch, S. Kulkarni, J. Lott, N. Suri, and A. Uszok. "KAoS semantic policy and domain services: An application of DAML to Web services-based grid architectures." Presented at the Proceedings of the AAMAS 03 Workshop on Web Services and Agent-Based Engineering, Melbourne, Australia, July, 2003.
23. Johnson, M., P. Feltovich, J.M. Bradshaw, and L. Bunch. *Coordinating Human-Robot Teams through Dynamic Regulation*, 2007.
24. Klein, G., P.J. Feltovich, J.M. Bradshaw, and D.D. Woods. "Common ground and coordination in joint activity." In *Organizational Simulation*, edited by W.B. Rouse and K.R. Boff, 139-184. New York City, NY: John Wiley, 2004.
25. Klein, G., D.D. Woods, J.M. Bradshaw, R. Hoffman, and P. Feltovich. "Ten challenges for making automation a "team player" in joint human-agent activity." *IEEE Intelligent Systems 19*, no. 6 (November-December 2004): 91-95.
26. Lott, J., J.M. Bradshaw, A. Uszok, and R. Jeffers. "Using KAoS policy and domain services within Cougaar." Presented at the Proceedings of the Open Cougaar Conference 2004, New York City, NY, 20 July, 2004, 89-95.
27. Malone, T.W. and K. Crowston. "What is coordination theory and how can it help design cooperative work systems?" Presented at the Conference on Computer-Supported Cooperative Work (CSCW '90), Los Angeles, CA, 7-10 October, 1990, 357-370.
28. Muscettola, N., P.P. Nayak, B. Pell, and B.C. Williams. "Remote Agent: To boldly go where no AI system has gone before." *Artificial Intelligence 103*, no. 1-2 (August 1998): 5-48.
29. Sierhuis, M., J.M. Bradshaw, A. Acquisti, R. Van Hoof, R. Jeffers, and A. Uszok. "Human-agent teamwork and adjustable autonomy in practice." Presented at the Proceedings of the Seventh International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS), Nara, Japan, 19-23 May, 2003.
30. Smith, W.J. *The Behavior of Communicating*. Cambridge, MA: Harvard University Press, 1977.
31. Tambe, M., W. Shen, M. Mataric, D.V. Pynadath, D. Goldberg, P.J. Modi, Z. Qiu, and B. Salemi. "Teamwork in cyberspace: Using TEAMCORE to make agents team-ready." Presented at the Proceedings of the AAAI Spring Symposium on Agents in Cyberspace, Menlo Park, CA 1999.
32. Uszok, A., J.M. Bradshaw, M. Johnson, R. Jeffers, A. Tate, J. Dalton, and S. Aitken. "KAoS policy management for semantic web services." *IEEE Intelligent Systems 19*, no. 4 (July/August 2004): 32-41.