Human-Robot Coordination through Dynamic Regulation

Matthew Johnson, Paul J. Feltovich, Jeffrey M. Bradshaw, and Larry Bunch

Abstract— **Several key aspects of coordination such as teamwork, roles, and communication, are enabled and driven by, and even largely defined by, various systems of regulation. One key feature of all these elements in human coordination is their dynamic nature. We have developed a framework to provide a dynamic regulatory system for supporting coordination in human-robot teamwork. This framework supports the definition and functions of roles within teams, as well as the creation of subteams and the roles within them. It also serves to regulate communications in support of coordination. We have demonstrated our system with a team of two humans and five robots performing advanced coordination while trying to apprehend an intruder hiding on a cluttered pier. This work lays the foundation for human-robot coordination based on dynamic regulation.**

I. INTRODUCTION

MANY things come to mind when talking about coordination; teamwork, roles, communication. What is the difference between coordination and teamwork? Tambe [1] suggests a common goal is what separates teamwork from mere coordination. Making the distinction between the two is not the goal of this paper; instead we focus on similarities that distinguish both of these concepts from independent activity. There is some behavior change that we identify as coordination or teamwork. Regardless of whether the participants have created a formal team, have well defined roles, or have explicitly conveyed their common goal, there is still something affecting behavior to enable coordination. For example, when checking out of a grocery store people tend to get in line, enter into well known scripted interaction with the check-out clerk, and so forth. Line formation and common scripts are examples of social regulatory devices people use to coordinate within everyday activity [2].

When a team is formed to perform some task, roles are frequently used to divide tasks among team members. For example, a football team will assign someone the roles of lineman and receiver. Being the lineman really doesn't mean much by itself, until one considers the regulatory structures, e.g., authorizations and obligations, associated with the role. For example, only the center lineman is authorized to snap the ball to start the play and is obligated to protect the quarterback. The receiver is obligated to run the specified route. It is these regulatory structures that largely define the role. Adherence to the regulatory system can result in success, and observers are likely to describe the activity as highly coordinated. Violations of the regulatory structures associated with the roles will likely result in confusion and a breakdown in coordination. Roles on teams may also be more informal or general, such as simply being a team "member" or "partner," but it is still the associated policies (implicit and/or explicit) that define the role and drive the behavior of the actor filling the role (e.g., the obligation to help rather than hinder a teammate as a "member" of a team). Roles are also used to form organizational structures, such as the hierarchical military chain of command. But again, however, the tree structure depicting such an organization on paper only takes on functional meaning when one considers the underlying obligations, rights, standards, traditions, and other regulatory considerations that make up the chain of command and drive the relational interactions. When robots participate on teams with humans, it is not enough that they just assume the title associated with a role or perform associated algorithmic behavior. It is important that they adhere to the regulatory structures that define that role.

Communication is often seen as a critical component of coordination. Indeed, without some communication, either verbal or non-verbal [3], explicit or implicit [4], it is hard to argue that there is any coordination at all. Even communication is managed by formal or informal regulatory systems. A formal example of a regulatory system that manages communication is found in the aviation community. The critical nature of aviation requires strict and well defined communications to ensure accuracy and brevity. An informal example is children in a class who are told to work together on a problem. They naturally know they must communicate, establish a common plan, set coordination points, and provide feedback and status updates.

The discussion so far advances the view that many key aspects of coordination such as teamwork, roles, and communication, are enabled and driven by, and even largely defined by, various systems of regulation. One key feature of all these elements in *human* coordination is their dynamic nature. Roles are reassigned, roles change, team members come and go, team structure changes, exceptions are made, obligations are waived, communication requirements are modified based on new context, and the regulatory systems must be adapted to meet the team's changing needs. This flexibility leads to robust human coordination and should be aspired to in human-robot teams.

Toward this end, we have developed a framework to provide a dynamic regulatory system for supporting coordination in human-robot teamwork. This framework supports the definition and functions of roles within teams, as well as the creation of subteams and the roles within

---

Manuscript received September 14, 2008. This work was supported in part by the Office of Naval Research under the Coordinated Operations project and also by the Army Research Lab's Team Performance grant.

M. Johnson, P. Feltovich, J. M. Bradshaw, and L. Bunch are with the Institute for Human and Machine Cognition, Pensacola, FL 32502 USA (phone: 850-202-4476; e-mail: mjohnson@ ihmc.us).

them. It also serves to regulate communications in support of coordination. We begin with brief review of related work. Next we present an abbreviated description of our regulatory framework. We then describe how we support informal regulation common in non-team based interaction and progressively move toward more formal and restrictive regulation typical in teams and other advanced organizations. Finally, we discuss a demonstration where we implemented many of the features discussed thus far in a single scenario with a team of two humans and five robots performing advanced coordination while trying to apprehend an intruder hiding on a cluttered pier.

II. RELATED WORK

There has been considerable work on robot-robot coordination [5][6][7][8]. The teams involved are typically fully autonomous or have a very minimal role for the human, with the interaction limited to simply displaying system state. Some teams use low level biologically inspired reactive behaviors [5] while others employ algorithmic solutions [6], mission planning software [7], or market-based negotiation [8]. Although there is much value in pursuing fully autonomous systems, it has been shown that some coordination with a human can improve performance [9]. It is also important to note that humans typically do not trust robots, alone, to perform critical tasks. Most mobile robots being used today are teleoperated by one or more humans [10]. There is also significant value to keeping humans in the loop and it is important to address this at design time [11]. There have been several examples of designing systems with the intention of keeping the human as a team player [9][12]. Most of these have focused on interaction with one robot, although a few have attempted two or three [13]. There has been even less work on including multiple humans, except for the common occurrence of allowing multiple people to teleoperate a single robot [14].

Our work differs from the previous work in several ways. First we insist on allowing humans to remain "in-the-loop" at any level of control they desire. We also allow multiple humans to participate on the team. Lastly, we provide a regulatory mechanism to help the robots coordinate with each other as well as humans in a manner that can enhance performance and safety. There has been a significant amount of theoretical work in this area, and we leverage this [15][16][17], as well as some robot-robot coordination based on this theoretical orientation [1].

III. REGULATORY FRAMEWORK (KAOS)

Since it was important that the regulatory system be applicable to a variety of robotic platforms and agents, we used KAoS [18]. It provides a general framework for regulation of a variety of systems including agent-based systems [19], web services, and grid services [20]. 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. All agents 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. To support heterogeneous robots involved in our project, we use the KAoS Robot extension to KAoS [21]. This provides a generic wrapper for each type of robot and a consistent interface for client systems to access the robots. KAoS Robot allows for control at various levels without restricting the control architecture. It also enables policy checking and enforcement for the robots, providing the awareness to handle the regulation required for coordination.

The main regulatory device used by KAoS is the policy. We define a policy as *an enforceable, well-specified constraint on the performance of a machine-executable action by a subject in a given situation* [19]. Policies are a means 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 is allowed (*positive authorizations* policies) or not allowed (*negative authorizations* policies) 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*).

KAoS polices are written in OWL (Web Ontology Language: http://www.w3.org/ 2004/OWL) and the details are explained in greater detail in [19]. The implementation details of the generic robot wrapper to enable application of the regulatory system are explained further in [21]. Armed with a regulatory system in the form of policies and an enforcement mechanism for heterogeneous robots, we now address coordination in a variety of contexts.

IV. VARIOUS DEGREES OF COORDINATION THROUGH REGULATION

A. Informal Coordination

First we consider the simplest (although not without nuance) coordination involving a single human and a single robot performing independent tasks in shared space. The coordination involved is basically managing the shared space. People have developed some customs and laws (two types of regulatory mechanism) to handle this type of interaction such as walking on the right side of a hallway or driving on the right side of the highway (at least in the U.S.). Robots might participate in these kinds of customs and may even adopt additional obligations such as warning neighboring people about impending movement to avoid both surprise and possible accidents.

We have demonstrated policies of this type in previous work [22], for example, by requiring robots to beep before moving. Although this simple task could be hard coded into the robotic behavior, using an external regulatory system provides more flexibility and control over the management of such behaviors. The policy can be applied across all robotic platforms without the need for implementation on each specific robot. The policy can also be more generally stated to "warn" before moving, allowing each robot use its own type of warning mechanism. Additional context can be applied; such as warn before moving, but only if someone is within five feet. For a fuller explanation of the advantages see [21].

B. More Formal Coordination

We now move to a slightly more complex example of coordination that still only involves a single person and a single robot, but now entails some joint activity. Here the coordination may involve simple task division (e.g. each member searches their own independent area) or more complex coordination in time or space (e.g. baton passing). This type of coordination typically requires more communication and is defined by its own set of regulations. For example, when working with someone on a joint task, it is expected that one partner would inform the other if a problem were encountered. If a robot is 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 this type of general teamwork policy as follows:

> *A Robot is obligated to notify the Requestor when requested Action is Finished (includes Completed, Aborted, and Failed)*

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

As stated in the introduction, dynamics is a crucial part of coordination in ongoing joint activity. KAoS provides a mechanism to support runtime addition and modification of policies in support of coordination. Although there are instances where predefined policies can be applied statically, there are also cases where policies need to be added or modified during the activity. Our framework therefore allows for both permanent and temporary changes to the regulations in force at any given time. Additionally, there are domain or context specific coordination devices that cannot be handled by generic policies. New policies can be created that are specifically tailored to the coordination desired. 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. People might simply state “Let me know when you see the target.” This phrase is the equivalent of establishing an obligation to inform the other party when specified conditions are met. We have created a similar 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.

C. Additional Peers

The addition of peers again changes the regulations that apply to members of a group. Some additional constraints that may apply include acknowledging requests. Acknowledgement is customary in some cases, but it also can have a functional role. Individuals may not be able to devote their full attention and may need acknowledgement to free them to perform some other tasks. We have implemented a policy that requires robots to acknowledge requests.

With more then one person creating policies (and even with just one,) it is likely that conflicting policies may be created. KAoS provides automatic de-confliction of policies, if it is possible. If de-confliction is not possible, KAoS will warn the policy creator at the time the conflicting policy is added to the system specifically identifying the policy in conflict. We demonstrate de-confliction with our acknowledgement policy. We wanted to model the fact that people do not always verbally acknowledge requests, particularly when they are directly observable. Direct observability means that when the requestor (e.g., the human) sends the communication to the receiver (the robot), 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 5 meters, and then can be seen starting to move forward, there is 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 it is possible to automatically de-conflict these policies and achieve the desired behavior.

D. Roles

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 dependant, for example the role “Team Leader” in the 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 some re-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.

These sets are not claimed to be complete, but lay the foundation for real world implementation of human-robot coordination, based on previous theoretical work [15][16][17] and simulation and robot-robot approaches [1] that demonstrated the feasibility and utility of such general coordination rules. It is expected that these sets will be revised and expanded.

E. Leaders

One interesting role is that of "Leader". 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:

<u>Chain of Command Policy Set</u>

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 involves notification for maintaining common ground among team members:

<u>Notification Policy Set</u>

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).*

Again these policy sets are viewed as a starting point and a test bed for the realistic implementations.

F. Organizational Structure

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.

V. DEMONSTRATION

A. *The Mission*

Figure 1 The pier

Consider a scenario in which you are asked to find an intruder hiding on a cluttered pier. To support your search, you are provided a team composed of an additional human and five robots. Your task is to coordinate your team to apprehend the target. This hypothetical scenario is one we demonstrated with real robots on the pier facility shown in Figure 1. 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. We also wanted to make sure the scenario included more than one human, since this provides its own challenges.

B. *The Team*

The available team members consisted of two humans and five robots (Figure 2). The humans were to play distinct roles. One was the overall "Commander" and in charge of establishing the teams and managing the overall process. The Commander operated remotely without direct sight of the area of operation through a combined speech and graphical interface. 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 along side the robots, usually in sight of them. He wore a backpack that carried a laptop to provide a similar interface as the Commander's, except through a head mounted display shown in Figure 2. The robot team members included four Pioneer 3AT robots with sonar and GPS. Two had pan-tilt-zoom cameras, one had a SICK laser, and one had a SICK laser and a camera. The fifth robot was a custom robot called the tBot. All the robots had onboard computers and used wireless routers for communication.

C. *Execution*

We used all of the previously discussed policy sets, including acknowledgement, progress appraisal, notification, and chain of command.

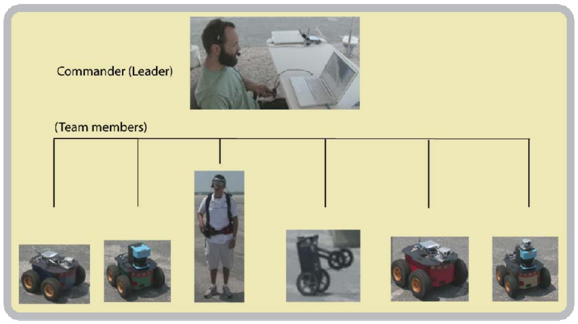

Figure 2 Initial two tiered team structure

The first logical subgoal in the overall task of apprehension is to make sure the intruder does not escape. The pier facility has a fence line on one side and water on the other, leaving two possible avenues for escape. Two subteams needed to be formed so each could be assigned to a side. The initial team configuration is shown in Figure 2. Using natural language, the Commander composed two teams and assigned their leaders. The resulting team structure is shown in 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 during team formation, since there was no external indication from the robots that the team assignment has occurred. 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. This is a normal coordination tool employed by humans. Once in position, the coordination policy took effect and the robot team leader reported. This policy is an excellent example of the type of coordination that cannot be handled through static mechanisms, but must be dynamic and flexible enough to support the current context.

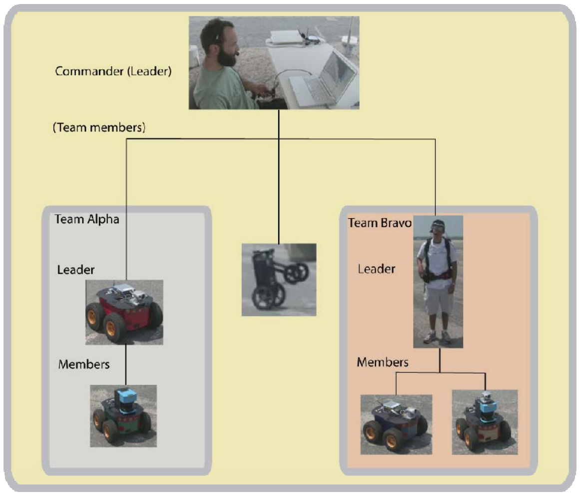

Figure 3 Three tier hierarchical team composed of two sub teams (tBot still on original team)

With the boundary 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 extant 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 correct chain of command to acquire permission. Then 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.

VI. CONCLUSION

We have advanced the view that many key aspects of coordination are enabled and driven by, and even largely defined by, various systems of regulation. We have also highlighted the importance of the dynamic nature of these elements in coordination. We have developed several generic teamwork policies that regulate different aspects of human-robotic coordination. The policies provide direction to establish and preserve common ground among both human and robotic team members, as well as helping to maintain organizational integrity. They are defined and enforced externally to any specific robot API, so that as new robots join, they automatically acquire all the teamwork intelligence possessed by the other robots. The policies are grouped into sets that address acknowledgement, progress appraisal, notification, and chain of command. We have developed a framework to provide a dynamic regulatory system for supporting coordination in human-robot teams and demonstrated this framework on real robots, using a multi-robot multi-human team, showing how teams, roles, and policies can be dynamically created, de-conflicted, and enforced at run-time.

REFERENCES

[1] Tambe, M.: Towards Flexible Teamwork. Journal of Artificial Intelligence Research (JAIR) 7: 83-124 (1997)

[2] Feltovich, P. J., Bradshaw, J. M., Clancey, W. J., & Johnson, M. (2007). In O'Hare, G., O'Grady, M., Ricci, A., & Dikenelli, O., Toward an ontology of regulation: Socially based support for coordination in human and machine joint activity. Engineering Societies for the Agents World VII. Lecture Notes in Computer Science Series. Heidelberg, Germany: Springer-Verlag.

[3] Feltovich, P.J., Bradshaw, J.M., Jeffers, R., Suri, N. & Uszok, A. (2004). Social order and adaptability in animal and human cultures as analogues for agent communities: Toward a policy-based approach. In A. Omacini, P. Petta, & J. Pitt (Eds.), Engineering societies for the agents world IV (pp.21-48). Lecture Notes in Computer Science Series. Heidelberg, Germany: Springer-Verlag.

[4] Parker, L. E., The Effect of Action Recognition and Robot Awareness in Cooperative Robotic Teams, Proceedings of the 1995 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), August 1995: Volume 1, pg. 212-219.

[5] Balch, T. and Arkin, R.C., Communication in Reactive Multiagent Robotic Systems, , Autonomous Robots, 1(1): 27-52, 1995.

[6] A. Howard, L.E. Parker, G.S. Sukhatme, "Experiment with a Large Heterogeneous Mobile Robot Team: Exploration, Mapping, Deployment and Detection," in International Journal of Robotics Research, vol. 2, no. 5-6, pp. 431-447, May-June 2006.

[7] P. Ulam, Y. Endo, A. Wagner, R. Arkin, "Integrated Mission Specifications and Task Allocation for Robot Teams – Design and Implementation," in IEEE International Conf. Robotics and Automation, 2007, pp. 4428-4435.

[8] M.B. Dias, et al, "Dynamically Formed Human-Robot Teams Performing Coordinated Tasks," in Proceedings of the AAAI Spring Symposium on Human Robot Teams, March 2006

[9] T. Fong, C. Thorpe, and C. Baur, Robot as Partner: Vehicle Teleoperation with Collaborative Control, Workshop on Multi-Robot Systems, Naval Research Laboratory, Washington, D.C. March 2002

[10] Burke, J.J., Murphy, R.R., Coovert, M., and Riddle, D., "Moonlight in Miami: A Field Study of Human-Robot Interaction in the Context of an Urban Search and Rescue Disaster Response Training Exercise," Human-Computer Interaction, special issue on Human-Robot Interaction, Vol. 19, Nos. 1-2, pp.85-116, 2004.

[11] Grosz, B., 1996. " Collaborative Systems: 1994 AAAI Presidential Address." 2(17), pp. 67 – 85

[12] C. Breazeal, A. Brooks, J. Gray, G. Hoffman, C. Kidd, H. Lee, J. Lieberman, A. Lockerd, and D. Mulanda (2004). "Humanoid Robots as Cooperative Partners for People." Submitted to IJHR (2004).

[13] Brennan Sellner, Frederik W. Heger, Laura M. Hiatt, Reid Simmons and Sanjiv Singh. Coordinated Multi-Agent Teams and Sliding Autonomy for Large-Scale Assembly. Proceedings of the IEEE, Vol. 94, No. 7, July 2006

[14] Ashley Tews, Maja J. Mataric, Gaurav S. Sukhatme: A scalable approach to human-robot interaction. ICRA 2003: 1665-1670

[15] Cohen, P. R., & Levesque, H. J. (1991). Teamwork. Technote 504. Menlo Park, CA: SRI International, March.

[16] Christofferson, K., & Woods, D. D. (2002). How to make automated systems team players. In E. Salas (Ed.), Advances in Human Performance and Cognitive Engineering Research, Vol. 2. JAI Press, Elsevier.

[17] Clark, H. H. (1996). Using Language. Cambridge, UK: Cambridge University Press.

[18] Bradshaw, J. M., et al. (2003). Representation and reasoning for DAML-based policy and domain services in KAoS and Nomads. Proceedings of the Autonomous Agents and Multi-Agent Systems Conference (AAMAS 2003). Melbourne, Australia, New York, NY: ACM Press,

[19] Bradshaw, J. M., et al. (2004). Making agents acceptable to people. In N. Zhong & J. Liu (Ed.), Intelligent Technologies for Information Analysis: Advances in Agents, Data Mining, and Statistical Learning. (pp. 355-400). Berlin: Springer Verlag.

[20] Johnson, M., Chang, P., Jeffers, R., Bradshaw, J. M., Soo, V.-W., Breedy, M. R., Bunch, L., Kulkarni, S., Lott, J., Suri, N., & Uszok, A. (2003). KAoS semantic policy and domain services: An application of DAML to Web services-based grid architectures. Proceedings of the AAMAS 03 Workshop on Web Services and Agent-Based Engineering. Melbourne, Australia,

[21] Johnson, M., Jeffrey M. Bradshaw, Paul Feltovich, Renia Jeffers, Hyuckchul Jung, and Andrzej Uszok, A Semantically Rich Policy Based Approach to Robot Control, Proceedings of the International Conference on Informatics in Control, Automation and Robotics, 2006

[22] Bradshaw, J. M., et al.(2005), Kaa: Policy-based Explorations of a Richer Model for Adjustable Autonomy, Proceedings of the International Joint Conference on Autonomous Agents and Multi Agent Systems, 2005