Human-Agent Teamwork and Adjustable Autonomy in Practice

Maarten Sierhuis[1], Jeffrey M. Bradshaw[2], Alessandro Acquisti[1], Ron van Hoof[3], Renia Jeffers[2], Andrzej Uszok[2]

1. Research Institute for Advanced Computer Science (RIACS)
NASA Ames Research Center, CA, USA

2. Institute for Human and Machine Cognition (IHMC)
University of West Florida, Pensacola, FL, USA

3. QSS Group, Inc.
NASA Ames Research Center, CA, USA

{msierhuis, aacquisti, rvanhoof}@mail.arc.nasa.gov
{jbradshaw, rjeffers, auszok}@ai.uwf.edu

keywords Agents, Teamwork, Adjustable Autonomy, Work Practice Modeling, Simulation

Abstract

This paper outlines a preliminary perspective on teamwork and adjustable autonomy in groups involving a mix of humans and autonomous agents. Unlike other approaches to agent teamwork, a human-centered perspective to human-agent interaction is used. The paper describes how we are integrating the Brahms and KAoS agent frameworks in order to model and simulate realistic work situations in space and to support the design of robotic and software agents for human-agent teamwork.

1. Introduction

Whereas early research on agent teamwork focused mainly on agent-agent interaction, there is a growing interest in various dimensions of human-agent interaction. Unlike some previous autonomous systems that were deliberately designed to take remote humans out of the loop, many new space efforts are specifically motivated by the need to support close human-agent interaction.[1] Under NASA sponsorship, we are investigating issues in human-robotic teamwork and adjustable autonomy. Future human missions to the Moon and to Mars will need more effective human-agent interaction. Astronauts will live, work, and perform laboratory experiments in collaboration with robots inside and outside their spacecraft and habitats on planetary surfaces.

In this paper, we outline a preliminary approach to adjustable autonomy and human-centered teamwork. Unlike other kinds of approaches, a human-centered perspective requires the design of agents to be problem-driven, activity-centered, and context-bound. Thus we must begin with a detailed understanding of how people actually work. To this end, researchers at NASA Ames have developed Brahms, an agent-based design tool that can be used to model and simulate realistic work situations in space. The ultimate objective is to produce agent-based simulations in Brahms that could form the basis for the design of robotic and software agents for actual operations. On its part, the Institute for Human and Machine Cognition (IHMC) is enhancing KAoS agent services to incorporate explicit general models of teamwork, mobility, and resource control appropriate for space operations scenarios. These models are represented largely in the form of policies.

In this paper we describe how Brahms and KAoS are being integrated into a single environment for developing human-agent work systems. We will describe how we are beginning to apply this approach to human-agent collaboration in space applications. Consistent with our emphasis on understanding teamwork in practice, the two projects described in this paper began with detailed studies of how astronauts and scientists actually work in space settings. While the first project deals with the work of geologists and biologists working on the surface of Mars, the second project deals with the daily work

[1] Recent research highlights the fact that even—or perhaps especially—highly autonomous systems have a strong requirement for effective interaction with people.

routine onboard the International Space Station (ISS).

2. Brahms

Brahms is an agent modeling, simulation and execution environment. Brahms has an agent-oriented language with a well-defined syntax and semantics. A Brahms model can be used to simulate human-machine systems, for what-if experiments, for training, "user models," or controlling intelligent assistants and robots [1, 2]. The run-time component—the Brahms virtual machine—either simulates agent and object behavior or else executes a Brahms model as part of a real-time system.

The Brahms architecture is organized around the following representational constructs:

Groups of groups containing
Agents who are located and have
Beliefs that lead them to engage in
Activities specified by
Workframes
Workframes in turn consist of
Preconditions of beliefs that lead to
Actions, consisting of
Communication Actions
Movement actions
Primitive Actions
Other composite activities
Consequences of new beliefs and facts
Thoughtframes that consist of
Preconditions and
Consequences

Physical objects are represented as entities whose states change within workframes and thoughtframes; conceptual objects represent human conceptualizations (e.g., the idea of an "experiment").

Brahms is based on the theory of situated action [3, 4]. The activity framework, which describes chronological behaviors, may be contrasted with the goal-driven framework in Soar and ACT-R, [5, 6], which functionally abstracts behavior in terms of tasks. Brahms offers to the researcher a tool to represent and study human behavior from the perspective of activity theory and "work practice" [7, 8]. A serious limitation of traditional task analysis is that it leaves out informal logistics, such as how environmental conditions are physically detected. Consider, for example, the fact that conventional medical expert systems do not model how physicians perform patient exams.

Without considering such factors, analysts cannot accurately model how work and information actually take place, and therefore they cannot adequately design software agents assist people in their work. For these purposes, we need a model that includes not only aspects of reasoning found in an information-processing model, but also aspects of geography, agent movement, and physical changes to the environment found in a multi-agent simulation—such as interruptions, coordination, and impasses. A model of work practice focuses on informal, circumstantial, and located behaviors by which synchronization among humans and machines occurs and allows the researcher to model to the degree possible the distinction in activity theory between motives, activities, and task-specific goals [1, 9].

3. KAoS

The increased intelligence afforded by software agents is both a boon and a danger. By their ability to operate independently without constant human supervision, they can perform tasks that would be impractical or impossible using traditional software applications. On the other hand, this additional autonomy, if unchecked, also has the potential of effecting severe damage in the case of buggy or malicious agents. Techniques and tools must be developed to assure that agents will always operate within the bounds of established behavioral constraints and will be continually responsive to human control. Moreover, the policies that regulate the behavior of agents should be continually adjusted so as to maximize their effectiveness in both human and computational environments. To this end, researchers at IHMC have developed KAoS.

KAoS is a collection of componentized agent services compatible with several popular agent frameworks, including Nomads [10], the DARPA CoABS Grid [11], the DARPA ALP/Ultra*Log Cougaar framework (http://www.cougaar.net), CORBA (http://www.omg.org), and Voyager (http://www.recursionsw.com/osi.asp). The adaptability of KAoS is due in large part to its pluggable infrastructure based on Sun's Java Agent Services (JAS) (http://www.java-agent.org). For a full description of KAoS, the reader is referred to [12-15].

KAoS domain services provide the capability for groups of agents to be structured into organizations of agent domains and subdomains to facilitate agent-agent collaboration and external policy administration. Domains may represent any sort of group imaginable, from potentially complex organizational structures to administrative units to dynamic task-oriented teams with continually changing membership. A given domain can extend across host boundaries and, conversely, multiple domains can exist concurrently on the same host. Domains may be nested indefinitely and, depending on whether policy allows, agents may become members of more than one domain at a time.

KAoS policy services allow for the specification, management, conflict resolution, and enforcement of policies within domains. Policies are DAML (DARPA Agent Markup Language) specifications that constrain the performance of some type of action by one or more actors in a given situation. The policy ontology distinguishes between authorizations (i.e., constraints that permit or forbid some action) and obligations (i.e., constraints that require some action to be performed, or else serve to waive such a requirement). Through various property restrictions in the action type, a given policy can be variously scoped, for example, either to individual agents, to agents of a given class, to agents belonging to a particular group, or to agents running in a given physical place or computational environment (e.g., host, VM).

4. Integration of KAoS and Brahms

The integration of Brahms with KAoS allows every Brahms agent to be represented within one or more KAoS domains. With this comes the benefit of a) distribution of Brahms agents over multiple machines in a network, b) allowing Brahms agents to interact with agents written for other platforms, and c) allowing the specification, deconfliction, and enforcement of KAoS policies on Brahms agents.

All these capabilities move Brahms toward its goal of becoming a full-fledged software agent development platform. Originally the Brahms environment was developed as a closed, non-distributed simulation system. All agents in Brahms run in a single Brahms simulation engine on top of a Java virtual machine (VM). The Brahms environment is principally used as a simulation tool for simulating organizations of people and machines. Recently, the Brahms simulation engine can also run as a multi-agent execution engine (a VM), in which Brahms agents run in parallel as separate Java threads. With this capability it now becomes possible to design multi-agent systems with Brahms and test them first in a Brahms simulation. After the design has been tested with a scenario-driven simulation, the system can be deployed as an operational agent system using the Brahms simulation engine in execution mode.

5. Mobile Agents Project

In the Mobile Agents project, we rely on KAoS services to help us deploy several agents running in multiple Brahms VM's on different mobile agent systems over a wide-area wireless network. We have developed a model-based, distributed architecture that integrates diverse components in a system designed for lunar and planetary surface operations. Entities modeled within the system include space suits, cameras, all-terrain vehicles, robotic assistants, crew members in a local habitat, and the mission support team [16]. Brahms software agents run on multiple mobile platforms (see Figure 1). These "mobile agents" interpret and transform available data to help people and robotic systems coordinate their actions to make operations more safe and efficient.[2] Each person or system that needs the support of the architecture has a "personal agent" that they interact with as an assistant or advisor.

The Brahms-based mobile agent architecture (MAA) uses a novel combination of agent types so the software agents may understand and facilitate communications between people and between system components (see Figure 1). A state-of-the-art spoken dialogue interface is integrated with Brahms models, supporting a speech-driven field observation record and rover command system. An interactive science data storage facility (the Science Organizer [18]) in the Mars habitat is integrated with Brahms running in the habitat, while connected over a wide-area wireless network with Brahms running on a robot, two ATV's and two space suits being worn by the EVA astronauts.

[2] Our use of the term to describe physical "mobile agents" is not to be confused with the concept of mobile software agents [17].

Each Brahms runtime environment has personal agents serving the physical mobile agent (e.g., the Habcom, the EVA astronaut, the ATV, the robot) An important aspect of the methodology involves first simulating the entire system in Brahms, then configuring the agents into a run-time system. Thus, Brahms provides a language, engine, and system builder's toolkit for specifying and implementing multi-agent systems.

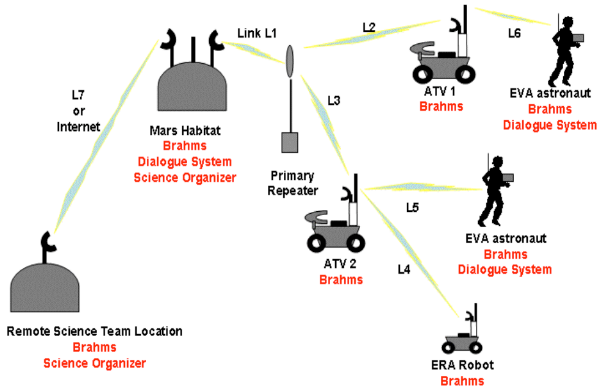

Figure 1. Mobile Agent Architecture

KAoS-Brahms integration addresses several issues. First, given that the systems and people are mobile, and that the wide-area network (5-10 km distance from hab to EVA team) is not robust, the software agents that control the physical entities need to be able to handle communication failures between agents running across the network. KAoS allows the Brahms software agents to communicate across the network, using KAoS' implementation of JAS and agent domains. Second, KAoS helps the agents to recover gracefully when communications among agents fails, due to problems in the network or problems with the receiver agent not responding.

The need for human-agent teamwork in the architecture comes from the type of support that the humans and robotic systems need in accomplishing their tasks. The task being researched in the Mobile Agent project is that of a planetary extra-vehicular activity (EVA) on Mars or some other planetary surface. We use the metaphor of a CapCom during the Apollo missions to help us think about the kind of agent support needed [19]. The CapCom was an astronaut in mission control who was the only person in direct communication with the crew on the Moon. The CapCom was a team member, collaborating with the crew performing tasks. The CapCom kept track of time, locations, placements of artifacts, and so forth. The crew used the CapCom as a sort of consultant when needed, while the CapCom acted as a director when they ran out of time, or couldn't find the location to which they were going. Our personal agents are designed to fill an analogous role. On a Mars EVA, CapCom's role is fulfilled by one of the crewmembers staying back in the habitat. We call this person the HabCom. HabCom's personal agent helps the HabCom and the EVA astronauts in tracking the EVA schedule. The HabCom's agent is constantly tracking the location of the EVA crew, as well as monitoring their health and the progress of their scheduled activities. When important thresholds are violated, the HabCom agent signals alarms to the HabCom and the EVA crew. It can also start and stop important EVA telemetry. At the same time, the EVA astronauts' personal agents support the EVA crew in storing, annotating and correlating gathered science data. It also helps the people to communicate with the ERA robot [20, 21] and other mobile agents.

KAoS policies allow us to specify the boundaries of the capabilities of the agents. The agents dynamically adjust their level of autonomy, based on their awareness of where the crew is located and where it is with regards to the schedule. The personal agents communicate with each other to develop mutual awareness of the EVA context (newly created locations, crew activities, etc). The crew can interact with their personal agents using a state-of-the-art speech dialogue system that is integrated with Brahms, allowing people to talk to their agents [22]. These personal agents mediate the crew's interactions with the robotic systems in the architecture.

6. Teamwork in Practice Project

In the Teamwork in Practice project we are investigating modeling of human-robot teamwork onboard the International Space Station (ISS). Consistent with our human-centered approach, we started with modeling the current work practice of the ISS astronauts onboard space station Alpha [23, 24]. A model of the ISS work practice is represented in a "day in the life" simulation. The model includes not only the execution of daily planned tasks, as scheduled in the onboard short-term plan (OSTP), and the crew's execution of these tasks according to the procedures, but it puts the execution of these tasks within the context of living and working onboard the ISS. The model includes a detailed conceptual model of the places and spaces of the ISS (the Geography Model). This allows for people, objects and artifacts to be located in a geographical space, which in turn allows the model to position the execution of tasks and activities of the crew-agents within the context of this space (e.g. crewmembers' desire to "fly" around the ISS to find the tools they need to accomplish the current task). Also, modeling work-life onboard the ISS means that the model needs to include sleeping, eating, personal hygiene and personal-time activities (e.g. reading a book after dinner), as well as important daily activities that are a) scheduled to include the ground (e.g. daily conference calls) and b) not scheduled but in the so-called "job jar" (i.e. a list of tasks the crew can work on whenever they have time).

Due to the constraints of the current ISS program, the crew of three spends much more of its time maintaining the station than originally planned. This creates a strain on the NASA ISS program, and has caused scrutiny of its actual research benefits, and consequently its funding. It is unlikely that, in the near future, the number of crew members onboard the ISS will increase. It is, therefore, desirable to investigate how autonomous robotic and software agents can help alleviate the crew in performing mundane or tedious tasks, so that they can spend their time more effectively.

Our research investigates human-robot teamwork from a human-centered perspective. This means that we do not hold the naïve objective of replacing humans with robotic systems. In contrast, we start from the human-centered view in which humans are seen as the crucial elements in the system, and robotic technology is fitted to serve human needs. For this, an effective approach to human-robotic teamwork is essential.

Modeling teamwork: Our policy-based teamwork model defines what constitutes a team and the nature of its collaborative activities. The set of policies we are designing for human-robotic interaction address traditional concerns such as authorization, access control, communication management, and resource control, but go beyond these in significant ways.

As long as the agent operates within the constraints specified as policy, it is otherwise free to act with complete autonomy. Policy-based constraints on behavior can be imposed and removed at any time. This coupling of autonomy with policy gives the agent maximum opportunity for local adaptation to unforeseen problems and opportunities, ensures effective human-agent interaction even for simple agents, and provides humans with assurance that behavior will be kept within desired bounds.

As one example, consider how policy can be used to ensure effective communication among team members. Previous research on generic teamwork models has explored this issue to a limited degree within the context of communication required to form, maintain, and abandon joint goals. However, more research is needed to address the complexities maintaining mutual awareness in human-agent as opposed to agent-agent interaction.[3] People need to understand what is happening—and why—when a teammate responds in a certain way; they need to be able to control the actions of an agent, even when the agent may not always wait for the human's input before making a move; and they need to be able to predict what will happen, even though the agent may alter its responses over time.

We are interested in incorporating insights from study of animal displays and nonverbal aspects of human behavior into robotic behavior. These context- and culturally-sensitive behaviors can be represented as KAoS-Brahms infrastructure policies that enable the appropriate expression to be made at the appropriate time without requiring each agent to individually encode that knowledge.

As another example, building on the work of [30], we are experimenting with KAoS notification policies to address several of these concerns. When an important event is signaled, the utility of various alternatives (e.g., notify the human, perform the action without interrupting the human, or do nothing) is evaluated. If a notification is required, the KAoS-Brahms infrastructure will take into account the current task and other contextual factors to perform the notification in a manner that is context-appropriate

[3] A detailed discussion of the principles and pitfalls of human-agent teamwork may be found in [29].

with respect to modality, urgency, and location of the human. Because the knowledge for effective human-agent interaction resides in the infrastructure rather than as part of the knowledge of each agent, agent development is simplified.

An important aspect of modeling teamwork is the notion of *joint goals*. Multi-agent teamwork research has typically held a simple view of joint goals [25-27]. However, Cartwright and Zander [28] point out the necessity of a more sophisticated view. They have found that the difficulties a team has in completing a task depend on the extent to which a clear goal is present, the degree to which the group goal mobilizes the energies of the members, the degree of agreement on how goals should direct activities of members, the degree of agreement about means to reach those goals, the degree to which members' activities are coordinated, and the availability to the group of various needed resources. To this extent, Cartwright and Zander define three levels of goals that are relevant in understanding the behavior of a team: individual goals; team goals, and individual goals for the team. We have extended this goal ontology with the concept of team goals for the individual (Table 1).

Modeling teamwork means that the agents need to be able to distinguish between these types of goals and the team members need to harmonize these goals into a coherent shared model of goals for the team. In other words, even though team members might have different individual goals for participating in the team, this does not necessarily mean that these goals will interfere with the shared team goals.

Table 1. Types of teamwork goals

Subject \ Focus	Individual	Team
Individual	Individual goal for self	Individual goal for the team
Team	Team goal as it applies to the individual	Team goal

We divide the team process into five general phases: 1) recognition of the need of help from other agents, 2) team formation, 3) ongoing coordination and team maintenance throughout task execution, 4) recognition of resolution or impasse, and 5) team disbanding. Throughout this process individual and team goals must be harmonized.

Adjustable autonomy for robotic systems: The goal of designing teamwork-supportive systems with adjustable autonomy is to make sure that for any given context the agents are operating at an optimal boundary between the initiative of the human and that of the agent (Figure 2). People want to maintain that boundary at the sweet spot in the tradeoff curve that minimizes their need to attend to interaction with the agent while providing them with a sufficient level of reassurance that nothing will go wrong. In principle, the actual adjustment of autonomy level could be performed either by a human, the agent itself, or some third party.

To the extent we can adjust agent autonomy with reasonable dynamism (ideally allowing fine-grained handoffs of control to occur "anytime") and with a useful range of levels, our teamwork mechanism can flexibly renegotiate roles and tasks among the human and robotic agents as needed when new opportunities arise or when breakdowns occur. It is important to note that the need for adjustments may cascade in complex fashion: interaction may be spread across many potentially distributed agents and humans who act in multiple-connected interaction loops. For this reason, in problems of realistic scale, adjustable autonomy may involve not merely a simple shift in roles among a human-agent pair, but rather the distribution of dynamic demands across many coordinated actors.

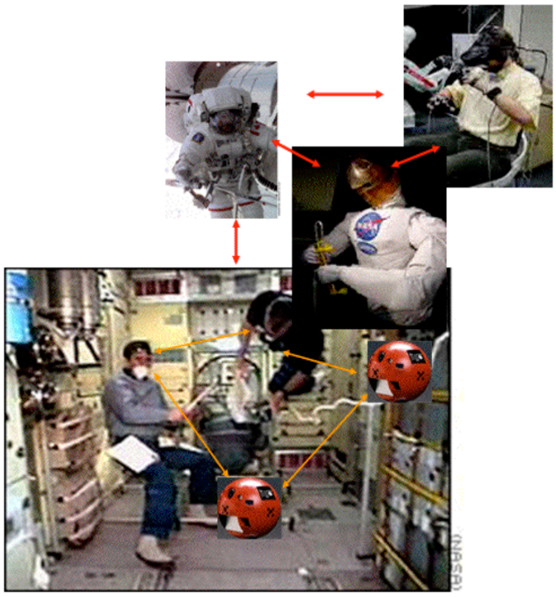

Figure 2. Human-Robot Teamwork on the ISS

We are developing KAoS policies that can be used to vary an agent's level of autonomy along several dimensions [29]. These include: 1) type or complexity of tasks or functions it is permitted to execute, 2) which of its functions or tasks may be autonomously controlled, 3) circumstances under which the agent will override manual control, 4) duration of autonomous operation, 5) the circumstances under which a human may be interrupted (or must be interrupted) in order to provide guidance [31].

7. Conclusion

In this paper we discuss a preliminary perspective on teamwork and adjustable autonomy. We discussed how we use Brahms and KAoS to implement a model of the work practice of human-robot teamwork, by focusing on the differences between people and autonomous agents. In particular, we discussed the integration of an agent simulation and development environment—Brahms—with a framework for distributed agent systems and teamwork policies—KAoS.

Currently we are using this KAoS-Brahms integrated environment in two NASA funded research projects; Mobile Agents and Teamwork in Practice. These projects were briefly discussed. Our current research results are preliminary but encouraging. Of course, many issues concerning the actual use of KAoS-Brahms remain to be explored. One such issue is the integration of broad and general KAoS teamwork policies and how they will impact the execution of activities for Brahms agents. We will address these in future papers as the work progresses.

Acknowledgements

This research is funded by the NASA IS HCC program. We would like to thank William J. Clancey, Mike Shafto, Rick Alena, John Dowding, Charis Kaskiris, Jeff Graham, Kim Schilcutt, Paul Feltovich and Niranjan Suri for their support and collaboration.

References

[1] M. Sierhuis, "Modeling and Simulating Work Practice; Brahms: A multiagent modeling and simulation language for work system analysis and design," *Ph.D. thesis, Social Science and Informatics*. Amsterdam, Netherlands: University of Amsterdam, 2001, pp. 350.

[2] W. J. Clancey, P. Sachs, M. Sierhuis, and R. van Hoof, "Brahms: Simulating practice for work systems design," *International Journal on Human-Computer Studies*, vol. 49, pp. 831-865, 1998.

[3] W. J. Clancey, "The Conceptual Nature of Knowledge, Situations, and Activity," in *Human and Machine Expertise in Context*, P. Feltovich, R. Hoffman, and K. Ford, Eds. Menlo Park, CA: The AAAI Press, 1997, pp. 247-291.

[4] L. A. Suchman, *Plans and Situated Action: The Problem of Human Machine Communication*. Cambridge, MA: Cambridge University Press, 1987.

[5] J. E. Laird, A. Newell, and P. S. Rosenbloom, "Soar: An architecture for general intelligence," *Artificial Intelligence*, vol. 33, pp. 1-64, 1987.

[6] J. R. Anderson and C. Lebiere, *The atomic components of thought*. Mahwah, NJ.: Lawrence Erlbaum Associates, 1998.

[7] M. Sierhuis, W. J. Clancey, C. Seah, J. P. Trimble, and M. H. Sims, "Modeling and Simulation for Mission Operations Work System Design," *Journal of Management Information Systems*, vol. Vol. 19, pp. pp. 85-129, in press.

[8] M. Sierhuis and W. J. Clancey, "Modeling and Simulating Work Practice: A human-centered method for work systems design," *IEEE Intelligent Systems*, vol. Volume 17(5), 2002.

[9] W. J. Clancey, "Simulating Activities: Relating Motives, Deliberation, and Attentive Coordination," *Cognitive Systems Research*, vol. 3, pp. 471-499, 2002.

[10] N. Suri, Bradshaw, J. M., Breedy, M. R., Groth, P. T., Hill, G. A., Jeffers, R., Mitrovich, T. R., Pouliot, B. R., & Smith, D. S, "NOMADS: Toward an environment for strong and safe agent mobility," presented at Proceedings of Autonomous Agents 2000, Barcelona, Spain, 2000.

[11] M. Kahn, Cicalese, C., "CoABS Grid Scalability Experiments," presented at Second International Workshop on Infrastructure for Scalable Multi-Agent Systems at the Fifth International Conference on Autonomous Agents, Montreal, CAN, 2001.

[12] J. M. Bradshaw, N. Suri, A. Cañas, R. Davis, K. M. Ford, H. R., R. Jeffers, and R. T., "Terraforming cyberspace," *IEEE Computer*, pp. 49-56, 2001.

[13] J. Bradshaw, M. Greaves, H. Holmbeck, W. Jansen, T. Karygiannis, B. Silverman, N. Suri, & A. Wong, "Agents for the masses: Is it possible to make development of sophisticated agents simple enough to be practical?," *IEEE Intelligent Systems*, 53-63, 1999.

[14] J. M. Bradshaw, S. Dutfield, P. Benoit, and J. D. Woolley, "KAoS: Toward an industrial-strength generic agent architecture," in *Software Agents*, J. M. Bradshaw, Ed. Cambridge, MA: AAAI Press/The MIT Press, 1997, pp. 375-418.

[15] Bradshaw, J. M., Uszok, A. Jeffers, R., Suri, N., Hayes, P., Burstein, M. H., Acquisti, A., Benyo, B., Breedy, M. R., Carvalho, M., Diller, D., Johnson, M., Kulkarni, S., Lott, J., Sierhuis, M. & Van Hoof, R. 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: ACM Press.

[16] W. J. Clancey, M. Sierhuis, C. Kaskiris, and R. v. Hoof, "Advantages of Brahms for Specifying and Implementing a Multiagent Human-Robotic Exploration System," presented at The 16th International FLAIRS Conference 2003, St.

Augustine, Fl., 2003.

[17] N. Vulkan, "Strategic design of mobile agents," *AI Magazine*, vol. 23, pp. 101-106, 2002.

[18] R. M. Keller, "Science Organizer: Information Sharing for Scientific Project Teams, <http://sciencedesk.arc.nasa.gov/organizer>," ScienceDesk project, NASA Ames, 2001.

[19] W. J. Clancey, "Roles for Agent Assistants in Field Science: Understanding Personal Projects and Collaboration," *IEEE Transactions on Systems, Man and Cybernetic*, Submitted.

[20] R. R. Burridge, J. Graham, K. Shillcutt, R. Hirsh, and D. Kortenkamp, "Experiments with an EVA Assistant Robot," presented at i-SAIRAS 2003, Nara, Japan, 2003.

[21] J. K. S. Graham, "Robot Tracking of Human Subjects in Field Environments," presented at i-SAIRAS 2003, Nara, Japan, 2003.

[22] J. Dowding, B. A. Hockey, J. M. Gawron, and C. Culy, "Practical Issues in Compiling Typed Unification Grammars for Speech Recognition," presented at 39th Annual Meeting of the Association for Computational Linguistics, Toulouse, France, 2001.

[23] M. Sierhuis, A. Acquisti, and W. J. Clancey, "Multiagent Plan Execution and Work Practice: Modeling plans and practices onboard the ISS," presented at 3rd International NASA Workshop on Planning and Scheduling for Space, Houston, TX, 2002.

[24] A. Acquisti, M. Sierhuis, W. J. Clancey, and J. M. Bradshaw, "Agent Based Modeling of Collaboration and Work Practices Onboard the International Space Station," presented at 11th Computer-Generated Forces and Behavior Representation Conference, Orlando, Fl., 2002.

[25] M. Tambe, "Towards Flexible Teamwork," *Journal of Artificial Intelligence Research*, vol. 7, pp. 83-124, 1997.

[26] M. P. Singh, "Intentions for Multiagent Systems," Microelectronics and Computer Technology Corporation, Austin, TX, MCC Technical Report Report Number KBNL-086-93, 1993.

[27] P. R. Cohen and H. J. Levesque, "Teamwork," *Special Issue on Cognitive Science and Artifical Intelligence, Noûs*, vol. 25, pp. 487-512, 1991.

[28] D. Cartwright and A. Zander, "Group Dynamics; Research and Theory," 3rd ed. New York, NY: Harper & Row, Publishers, 1968.

[29] Bradshaw, J., Sierhuis, M., Acquisti, A., Feltovich, P., Hoffman, R., Jeffers, R., Prescott, D., Suri, N., Uszok, A., and Van Hoof, R. (2002). Adjustable autonomy and human-agent teamwork in practice: An interim report on space applications. In Henry Hexmoor, Rino Falcone, and Cristiano Castelfranchi (Eds.), *Agent Autonomy*, Dordrecht, The Netherlands: Kluwer, in press.

[30] Schreckenghost, D., Martin, C., & Thronesbery, C. (2003). Specifying organizational policies and individual preferences for human-software interaction. Submitted for publication.

[31] G. Dorais, R. P. Bonasso, D. Kortenkamp, B. Pell, and D. Schrekenghost, "Adjustable autonomy for human-centered autonomous systems on Mars," presented at Proceedings of the AAAI Spring Symposium on Agents with Adjustable Autonomy, Stanford University, CA, 1999.