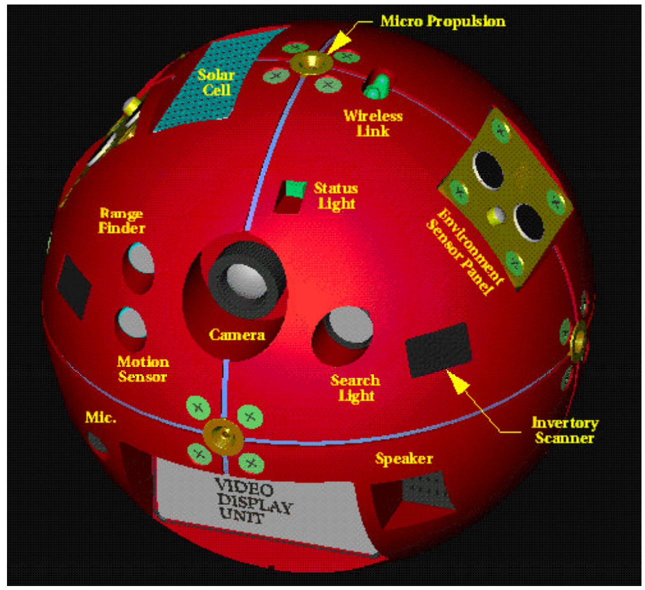
Adjustable Autonomy and Teamwork for the Personal Satellite Assistant

J. M. Bradshaw[3], M. Sierhuis[1], Y. Gawdiak[1], R. Jeffers[3], N. Suri[3], M. Greaves[2]

1. RIACS and NASA Ames, MS 19-39, Moffett Field, CA 94035, {msierhuis, ygawdiak}@mail.arc.nasa.gov
2. The Boeing Company, P.O. Box 3707, MS 7L-44, Seattle, WA 98124, mark.t.greaves@boeing.com
3. Institute for Human & Machine Cognition, UWF, 40 S. Alcaniz, Pensacola, FL 32501 {jbradshaw, rjeffers, nsuri}@ai.uwf.edu

Abstract

The Personal Satellite Assistant (PSA) is a softball-sized flying robot designed to operate autonomously onboard manned spacecraft in pressurized micro-gravity environments. We describe how the Brahms multi-agent modeling and simulation environment in conjunction with a KAoS agent adjustable autonomy and teamwork approach will be used to support human-centered design and operation of the PSA.



Introduction

The autonomous space systems of the future will need to perform many tasks involving close to real-time cooperation with people and with other autonomous systems. While these heterogeneous cooperating entities may operate at different levels of sophistication and with dynamically varying degrees of autonomy, they will require some common means of representing and appropriately participating in joint tasks. Equally important, developers of such systems will need tools and methodologies to assure that such systems will work together reliably, even when they are designed independently.

One example of such a system is the Personal Satellite Assistant (PSA), a softball-sized flying robot designed to operate onboard spacecraft in pressurized micro-gravity environments (figure 1) [9]. The PSA will incorporate environmental sensors for gas, temperature, and fire detection, providing the ability for the PSA to monitor spacecraft, payload and crew conditions. Video and audio interfaces will support for navigation, remote monitoring, and video-conferencing. Ducted fans will provide propulsion and batteries will provide portable power.

As an example of how the PSA might be used on future manned space missions, consider the following scenario, which emphasizes the collaborative aspects of human-robotic interaction:

A crewmember is awoken by a PSA at the requested time. The astronaut asks for a video briefing on the latest events, schedule changes, and priorities while she washes, and eats breakfast. The PSA follows the crewmember through her routine while giving the updates and then checks the inventory database to ensure that the necessary resources are available for the astronaut's first scheduled task. The crewmember logs into her homepage and sets several notifications to be programmed into the PSA to remind her of important activities and times for today's tasks. As the crewmember works at a payload rack the PSA tracks her movements and provides a remote data terminal capability to allow her to check on procedures and training instructions, and to support remote video-conferencing and email exchanges with remote colleagues. Later the crewmember conducts a delicate investigation in the glove-box. She requests support from the Principal Investigator (PI) on earth to help her walk through the procedure. The PI calls up a second PSA and maneuvers about the astronaut and glove-box to obtain an optimum view of the operation and to provide real-time feedback to the crewmember. Since the crewmember and the remote PI are absorbed in performing their tasks, the PSA's coordinate the details of their flight and their participation in joint and individual activities themselves, without requiring constant attention from their human partners. Moreover, the PSA's are not just passively waiting to be told what to do. They are actively looking for ways to be helpful to the humans in their current task as well as in ongoing responsibilities that have been delegated. For example, as the crewmember uses up supplies the PSA tracks the inventory tags and updates the inventory database. During a video inspection, a PSA notices that specimens in *habitat holding units need food. That evening a pair of PSA's use special integrated payload interfaces and cargo packages to inject supplies such as food into experimental units. One PSA injects the supplies and another collaborating PSA acts as a supply cargo carrier.*

While the interactions portrayed in the scenario seem simple and natural, experienced researchers in collaborative robotics will realize how many theoretical and practical issues this scenario raises. Because of the complications involved in such situations, the bulk of research in autonomy has naturally shied away from situations involving rich real-time interaction among a mixed group of human and artificial agents. But resolution of these issues cannot be postponed indefinitely if we truly are committed to a permanent joint human and robotic presence in space.

Although we currently envision the Personal Satellite Assistant (PSA) as the most accessible and practical initial testbed for our prototyping work in the design of collaborative robots, we are confident that our results will generalize to future cooperative autonomous systems of many other sorts. For instance, future human missions to the Moon and to Mars will undoubtedly need the increased capabilities for human-robot collaborations we envision. Astronauts will live, work, and perform laboratory experiments in collaboration with robots not only inside, but also outside the habitat on planetary surfaces. Specific examples of robots requiring close interaction with humans include Astronaut-Rover Interaction for Planetary Surface Exploration (ASRO) and the Marsokhod Planetary Rover (http:// img.arc.nasa.gov/ marsokhod/ marsokhod.html) and the Extravehicular Robotic Camera (AERCam) (http:// www.ri.cmu.edu/ projects/ project_311.html).

Methodologies and tools for design and implementation of human-centered approaches for cooperative autonomous systems are currently in their infancy. In this effort, we aim to combine the talents of members of our research team to develop theory and tools necessary for supporting "design to implementation" prototypes for the PSA and space systems with similar requirements.

PSA Motivation and Basic Capabilities

Enhancing the crew's ability to perform their duties is critical for successful, productive, and safe space operations aboard the Space Shuttle, Space Station, and during future space exploration missions to the Moon and Mars. Crew time on such missions is a precious resource and may cost hundreds of dollars per minute per astronaut. The limited number of crew members are required to maintain complex systems, assist with life-critical environmental health monitoring and regulation, perform dozens of major simultaneous payload experiments, and perform general housekeeping. As one example, consider the challenges of Shuttle Mission 89's flight on February 2, 1998:

"*One* astronaut, Andy Thomas, will undertake *several hundred* research runs involving *26* different science projects in *five* disciplines. The projects are provided by *33* principal investigators from the U.S., Canada, Germany and the U.K."

Safety considerations and size constraints are also important issues for many manned mission activities. Consider the "jungle of cables, power lines, air ducts, and drag lines obstruct[ing the] hatchway between Mir modules" (figure 2). Even if it were physically possible for an astronaut to enter congested spacecraft areas, protruding debris or other environmental hazards of one kind or another could pose serious safety risks.

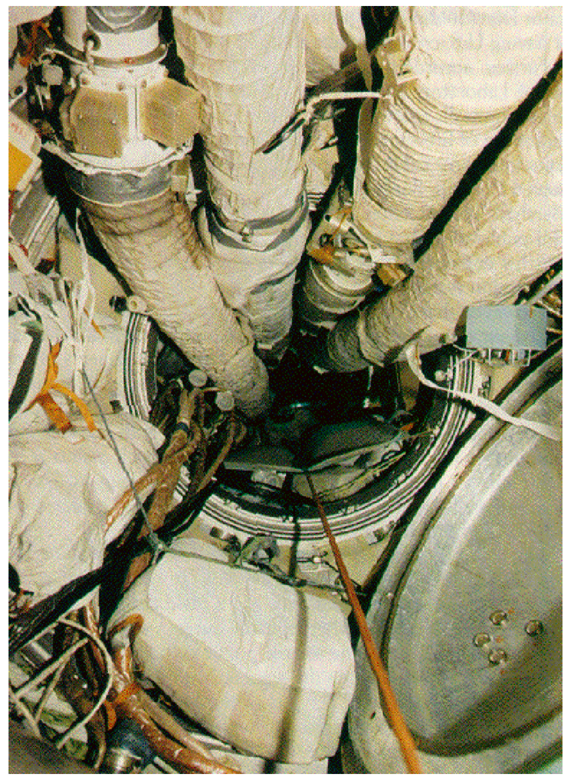


Figure 2. Obstructed hatchway between Mir modules

To function as an effective autonomous robot or semi-autonomous assistant, the PSA must first possess some basic foundational capabilities.

Navigation and control. The PSA must be capable of superb navigation and control. While at first glance control of such a device in a confined weightless environment may seem straightforward, this is not the case. Due to the presence of humans and sensitive micro-gravity experiments, it is critical that the PSA be able to move in a controlled fashion that assures that collisions will not occur. In a frictionless environment, velocity can increase rapidly. Holding a stationary position will require the development of active control technologies that can take into account the many influences that may be exerted on the PSA.

Sensing. The PSA must be able to observe its environment. It will function as an active super-sensor within a potentially under-sensed environment. Because of its small size and mobility, it will be able to make observations in places that are inaccessible to humans and validate information obtained from the fixed sensor suite.

Wireless communication. A wireless network will provide communication with spacecraft, ground operations, and remote crew operations. The wireless network will also connect the PSA to the spacecraft's avionics data and payload networks, and provide access to a system server that will provide off-PSA processing for computationally intensive tasks. Optimal distribution of computing tasks among the various processors can be maintained by packaging code as mobile agents [11; 19].

Diagnostics. The PSA must be capable of performing a broad range of diagnostic tasks from intelligent performance support for humans performing diagnostic tasks to more ambitious forms of automated diagnosis. Unfortunately, we do not currently have the resources to tackle the development of the detailed models of the space station required for sophisticated diagnosis. However we are collaborating with the Mission Operations Directorate at NASA Johnson Space Center to explore how they can use more sophisticated diagnosis techniques to assist the Station Duty Officer (SDO) in station monitoring. If this work is successful, we hope to use the resulting models in a future PSA prototype capable of providing sophisticated diagnostic assistance to the SDO, helping to eliminate ambiguities and validate hypotheses about space station anomalies [21].

Human interface. The PSA must support a variety of interfaces for the humans that interact with it. These include a remote data terminal, videoconferencing facilities, payload and maintenance procedure aids, just-in-time training, and various personal assistants providing task performance support. Given that hands-free operation will be the only form of interaction, speech understanding is a must.

Technical Approach

A human-centered approach to design requires first and foremost a thorough understanding of the kinds of interactive contexts in which humans and autonomous systems will cooperate. We have begun to investigate the use of Brahms [6] as an agent-based design toolkit to model and simulate behaviors of two or more PSA's with sets of crew members and ground controllers. The agent-based simulation in Brahms will eventually become the basis for the design of PSA functions for actual operations. On its part, IHMC and Boeing are enhancing their KAoS agent framework [1; 3] to incorporate an explicit general model of teamwork appropriate for space operations scenarios.

Brahms: An environment for multi-agent modeling and simulation

As part of a new effort, we will evaluate whether a model of human-robot collaboration in Brahms can be used not only as a design tool to understand human-robotic interaction, but also in conjunction with agents in the execution environment. Unlike traditional approaches to autonomous system design, our human-centered approach will base the design of the robotic agents on a real-world understanding of how the astronauts actually work and collaborate on the space station. Through crew interviews and observation we will develop a model of the work practice of the crew in various PSA use scenarios. Through the development of a multi-agent work practice simulation model, we will discover how the PSA can best collaborate with human team members while taking the systems and artifacts in its environment into account.

Theoretical foundations. A traditional task or functional analysis of work leaves out the logistics, especially how environmental conditions come to be detected and how problems are resolved. Without consideration of these factors, we cannot accurately model how work and information actually flows, nor can we properly design software agents that help automate human tasks or interact with people as their collaborators. What is wanted is a model that includes aspects of reasoning found in an information-processing model, plus aspects of geography, agent movement, and physical changes to the environment found in a multi-agent simulation. A model of work practice focuses on informal, circumstantial, and located behaviors by which *synchronization* occurs, such that the task contributions of humans and machines flow together to accomplish goals.

Our approach relates knowledge-based models of cognition (e.g., task models) with discrete simulation and the behavior-based subsumption architecture [5]. Agents' behaviors are organized into activities, inherited from groups to which agents belong. Most importantly, activities locate behaviors of people and their tools in time and space, such that resource availability and informal human participation can be taken into account.

A model of activities doesn't necessarily describe the intricate details of reasoning or calculation, but instead captures aspects of the social-physical context in which reasoning occurs [4]. Thus Brahms differs from other multi-agent systems by incorporating the following:

- Chronological activities of multiple agents;
- Conversations;
- Descriptions of how information is represented, transformed, reinterpreted in various physical modalities.

A Brahms model can be used to simulate human-machine systems for what-if experiments, for training, for "user models," or for driving intelligent assistants and robots. Brahms models are written in an *Agent-Oriented Language* (AOL) that has a well-defined syntax and semantics. The run-time component—the simulation engine—can execute a Brahms model; also referred to as a simulation run.

The architecture includes the following (simplified) 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

In addition, *active physical objects* (e.g., cameras, telephones, laptop computers) are modeled as entities whose state changes by Factframes. *Conceptual objects* are entities people have beliefs about, but that have no specific location (e.g., a mission) and are associated with physical objects (e.g., a particular orbiter).

Addressing teamwork issues in the KAoS agent framework

Given the mission scenarios and foundational capabilities described above, requirements for an agent architecture appropriate to the PSA begin to come into focus. Though we have thus far described the PSA casually as being autonomous, it is clear that it must support a spectrum of levels of autonomy, from highly-directed external control to significant self-directed activity *(adjustable autonomy)* [8]. Additionally, the PSA agent architecture must take into account not only its own goals but also reason about its commitments to take joint action with other agents, be they human or robotic *(teamwork)*. Though various theoretical approaches to multi-agent teamwork have appeared in the literature (e.g., [7; 20]), their claims have not yet been adequately evaluated in intensive real-time settings involving combinations of people and operational systems with significant autonomy. The use of Brahms design and simulation tools in conjunction with KAoS' theory-based multi-agent execution framework will help us better understand how teamwork happens in actual practice, and assure that implementation of autonomous cooperating systems are principled in design and reliable in operation.

Adjustable autonomy. One key challenge will be to allow dynamic control of the level of autonomy in PSA. Many autonomous systems are designed with fixed assumptions about what level of autonomy is appropriate to their tasks. They execute their instructions without taking into account that fact that the optimal level of autonomy may vary by task and over time, or that unforeseen events may prompt a need for either the human or the system to take more control. A system's level of autonomy can be varied along several dimensions such as: 1) type or complexity of the commands it is permitted to execute, 2) which of its subsystems may be autonomously controlled, 3) circumstances under which the system will override manual control (e.g., if a human operator is about to navigate the PSA into a wall), and 4) duration of autonomous operation.

The goal of designing systems with adjustable autonomy is to make sure that for any given situation and task the system is operating at the correct boundary between the initiative of the user and that of the system. People want to maintain that boundary at the sweet spot in the tradeoff curve that minimizes their need to attend to interaction with the system [10] while providing them a sufficient level of comfort that nothing will go wrong [16]. The actual adjustment of autonomy level can be performed by a person or a program, or by the agent itself. A variety of experiments will need to be conducted to understand the mechanisms and dimensions of adjustable autonomy best suited to the PSA.

Teamwork in mixed human-robotic environments. One of the hallmarks of the PSA scenario is that the PSA cooperates with the astronauts it interacts with, the other PSAs it might encounter, and even the space station equipment and experiments. At minimum, cooperation entails that a group of entities act in a coordinated fashion. However, we envision a much stronger type of cooperation for the PSA. Beyond merely acting in a coordinated way (as do, for example, cars on a road obeying the rules of the road), we would like the PSA to be able to implicitly and explicitly form teams with other agents that are based around shared goals. True teamwork is demanding: when the PSA teams with another PSA or the astronaut, the PSA must commit the resources required by the team, forego opportunities that are inconsistent with the team goals, persistently keep its relevant team goals and subgoals, and accept the overhead of forming, maintaining, and disbanding the team. However, the benefits of teamwork (robustness under unreliable actions and changing circumstance, multi-layered and distributed commitments to the shared goal) are critical to the type of behavior we would like to see the PSA exhibit.

The key concept in the theory of teamwork is that of a joint intention, which functions as the glue that binds team members together. The concept is formulated as a joint commitment to perform a collective action while in a certain shared mental state. By virtue of a largely-reusable explicit formal model of shared intentions, general responsibilities and commitments that team members have to each other are managed in a coherent fashion that facilitates recovery when unanticipated problems arise. For example, a common occurrence in joint action is when one team member fails and can no longer perform in its role. The general teamwork model entails as a formal consequence that each team member will be notified under appropriate conditions of the failure, and so does not require special-purpose exception handling mechanisms to do this for each possible failure mode.

Unfortunately, the power of a general-purpose teamwork model comes at a high price. Joint intention theory is built on an extremely powerful logical framework that includes explicit representation of mental attitudes like belief, goal, intention, and so forth. These attitudes are modeled in the traditional way: as new modal operators in a quantified modal logic. Hence, while the most general form of joint intention theory is representationally very attractive, it is computationally intractable. This tension between expressivity and computability is not limited to teamwork theories; in fact, it is a hallmark of all mentalistic theories of agent behavior and speech-act based agent communication. Thus, when designing agents which include strong teamwork assumptions and powerful communication languages (as do the PSA and other robots), it is critically important to reduce the power of these general models in a way that is sensitive to the agent's domain and expected range of action.

We will base our work on the PSA's agent-based teamwork capabilities on our research in multi-agent communication, collaboration, and information access developed in KAoS as part of the NASA-sponsored Aviation Extranet project [1; 2; 3]. By using the analysis and simulation capability in Brahms, we will be able to incorporate models of the PSA work environment and practices in our decisions about how to strategically weaken general joint intention theory without compromising the PSA's ability to perform in its environment. In this way, we will balance empirical analysis, simulation, and top-down theoretical considerations in arriving at a teamwork theory that will allow the PSA to meet the scenario goals. Teams will be formed, maintained, and disbanded through the process of agent-to-agent communication using an appropriate semantics. Agents representing various team members, from humans to autonomous systems to simple devices and sensors, will assure coherence in the adoption and discharge of team commitments and will encapsulate state information associated with each entity. Ongoing research is underway to allow heterogeneous agents of widely varying degrees of sophistication to be accommodated as team members [3]. Agent conversation policies are being designed to assure robust behavior and to keep computational overhead for team maintenance to an absolute minimum [12; 13; 14; 15; 18].

Status

Custom hardware components for the PSA have been fabricated including a custom air bearing assembly to float the PSA on an air table (figure 3).

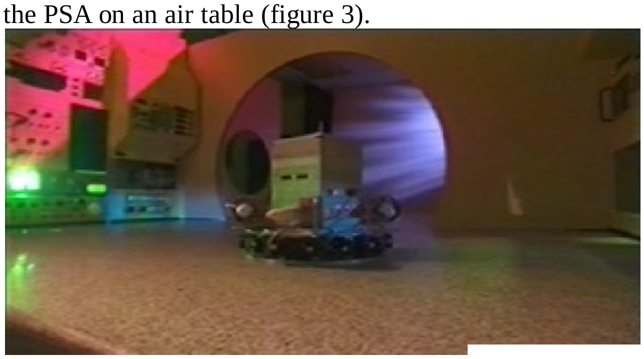


Figure 3. PSA Prototype on air table.

Onboard software to control attitude, and move the PSA prototype from point to point on the air table has been completed. A high-level, reactive execution language to specify and requests tasks to be performed by the PSA has been designed, as well as an initial speech interaction feasibility prototype [17]. A software simulation of the PSA has been developed using the Hybrid Concurrent Constraint (HCC) programming language in order to demonstrate goal-directed, reactive execution. We have performed a small experiment with Brahms to determine its suitability for modeling the PSA and its behavior within the space station (including interaction with the astronaut and sensors, as well as the movement through space). Figure 4 shows the graphical output of the simulation. The (blue) arrows show the communication between the astronaut and the PSA.

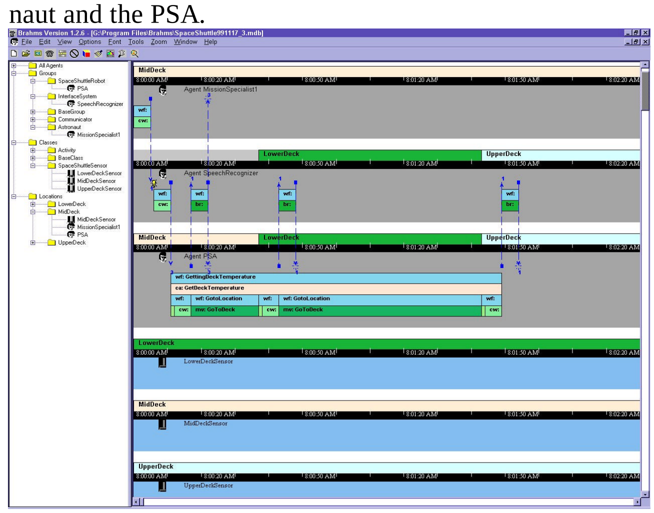


Figure 4. The 2D graphical output of a Brahms simulation of human interaction with the PSA and object in its environment.

The KAoS agent framework is being enhanced to support the PSA's more demanding requirements for teamwork, mobility, and conversation and resource management.

Conclusion

We are excited about the potential of the PSA as a platform for evaluating innovative hardware designs and intelligent software coupled to allow the flying robot to work independently or as a teammate with agents of all kinds and sophistication. The size and relatively small cost of the PSA makes it a more practical platform for trying out high-risk technologies than its full-sized satellite cousins. Especially intriguing is the prospect of agent architecture based on empirically-derived models, and incorporating adjustable autonomy and teamwork that are necessary to support reactivity to complex events in real time and a high level of interactivity with people.

Acknowledgements. We acknowledge the many contributions of the other members of PSA-related projects including, among others: Alessandro Acquisti, Maggie Breedy, Bill Clancey, Dan Clancy, Greg Dorais, Ken Ford, Vineet Gupta, Beth Hockey, Robert Hoffman, Frankie James, John Loch, James Lott, Debbie Prescott, Mike Shafto, Mark Sibenac, Al Underbrink, Ron van Hoof, and Brian Williams.

References

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

[2] Bradshaw, J. M., Gawdiak, Y., Cañas, A., Carpenter, R., Chen, J., Cranfill, R., Gibson, J., Hubbard, K., Jeffers, R., Mathé, N., Poblete, L., Robinson, T., Sun, A., Suri, N., Wolfe, S., & Bichindaritz, I. (1998). Toward an intelligent aviation extranet. G. Boy, C. Graeber, & J.-M. Robert (Ed.), *Proceedings of the International Conference on Human-Computer Interaction in Aeronautics (HCI-Aero 98),* (pp. 171-176). Montreal, Quebec, Canada, , Éditions de l'École Polytechnique de Montréal ,

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

[4] Chen, J. R., & Mathé, N. (1995). *Learning subjective relevance to facilitate information access.* NASA Ames Research Center, November.

[5] Clancey, W. J. (1997). *Situated Cognition: On Human Knowledge and Computer Representations.*, Cambridge: Cambridge University Press.

[6] Clancey, W. J., Sachs, P., Sierhuis, M., & van Hoof, R. (1998). Brahms: Simulating practice for work systems design. *International Journal of Human-Computer Studies*, 49, 831-865.

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

[8] Dorais, G., Bonasso, R. P., Kortenkamp, D., Pell, B., & Schrekenghost, D. (1999). Adjustable autonomy for human-centered autonomous systems on Mars. *Proceedings of the AAAI Spring Symposium on Agents with Adjustable Autonomy. AAAI Technical Report SS-99-06,* . Menlo Park, CA, , Menlo Park, CA: AAAI Press ,

[9] Gawdiak, Y., Bradshaw, J. M., Williams, B., & Thomas, H. (2000). R2D2 in a softball: The Personal Satellite Assistant. H. Lieberman (Ed.), *Proceedings of the ACM Conference on Intelligent User Interfaces (IUI 2000),* (pp. 125-128). New Orleans, LA, , New York: ACM Press ,

[10] Gershenfeld, N. A. (1999). *When Things Start to Think.*, New York: Henry Holt and Company.

[11] Gray, R., Cybenko, G., Kotz, D., & Rus, D. (2000). Mobile agents: Motivations, state-of-the-art systems, and frontiers. In J. M. Bradshaw (Ed.), *Handbook of Agent Technology*. (pp. in press). Cambridge, MA: AAAI Press/The MIT Press.

[12] Greaves, M., Holmback, H., & Bradshaw, J. M. (1999). What is a conversation policy? M. Greaves & J. M. Bradshaw (Ed.), *Proceedings of the Autonomous Agents '99 Workshop on Specifying and Implementing Conversation Policies,* (pp. 1-9). Seattle, WA, , ,

[13] Greaves, M., Holmback, H., & Bradshaw, J. M. (2000). Agent conversation policies. In J. M. Bradshaw (Ed.), *Handbook of Agent Technology*. (pp. in preparation). Cambridge, MA: AAAI Press/The MIT Press.

[14] Holmback, H., Greaves, M., & Bradshaw, J. M. (1999). A pragmatic principle for agent communication. J. M. Bradshaw, O. Etzioni, & J. Mueller (Ed.), *Proceedings of Autonomous Agents '99,* (pp. 368-369). Seattle, WA, , New York: ACM Press ,

[15] Holmback, H., Jeffers, R., Greaves, M., & Bradshaw, J. M. (2000). Composing and elaborating agent conversation policies. *Proceedings of the Agents 2000 Agent Communication Workshop,* . Barcelona, Spain, , ,

[16] Norman, D. A. (1997). How might people interact with agents? In J. M. Bradshaw (Ed.), *Software Agents.* (pp. 49-55). Cambridge, MA: The AAAI Press/The MIT Press.

[17] Rayner, M., Hockey, B. A., & James, F. (2000). Turning speech into scripts. *Proceedings of the AAAI Spring Symposium on Natural Language Dialogue with Robots,* . Menlo Park, CA: Stanford University, , ,

[18] Smith, I. A., Cohen, P. R., Bradshaw, J. M., Greaves, M., & Holmback, H. (1998). Designing conversation policies using joint intention theory. *Proceedings of the Third International Conference on Multi-Agent Systems (ICMAS-98),* (pp. 269-276). Paris, France, , Los Alamitos, CA: IEEE Computer Society ,

[19] Suri, N., Bradshaw, J. M., Breedy, M. R., Groth, P. T., Hill, G. A., Jeffers, R., Mitrovich, T. R., Pouliot, B. R., & Smith, D. S. (2000). NOMADS: Toward an environment for strong and safe agent mobility. *Proceedings of Autonomous Agents '99,* . Barcelona, Spain, , New York: ACM Press ,

[20] Tambe, M., Shen, W., Mataric, M., Pynadath, D. V., Goldberg, D., Modi, P. J., Qiu, Z., & Salemi, B. (19999). Teamwork in cyberspace: Using TEAMCORE to make agents team-ready. *Proceedings of the AAAI Spring Symposium on Agents in Cyberspace,* . Menlo Park, CA, , Menlo Park, CA: The AAAI Press ,

[21] Williams, B., & Gupta, V. (1999). Unifying model-based and reactive programming within a model-based executive. *Workshop on Principles of Diagnosis,* . , , ,