

R2D2 in a Softball: The Portable Satellite Assistant

Yuri Gawdiak¹, Jeff Bradshaw², Brian Williams³, Hans Thomas¹

1. NASA Ames, MS 269-4, Moffett Field, CA 94035, ygawdiak@mail.arc.nasa.gov, hans@artemis.arc.nasa.gov

2. The Boeing Company, P.O. Box 3707, MS 7L-44, Seattle, WA 98124, jeffrey.m.bradshaw@boeing.com

3. MIT Dept of Aeronautics and Astronautics, Cambridge, MA 02139, williams@mit.edu

ABSTRACT

The Portable Satellite Assistant (PSA) is a softball-sized flying robot designed to operate autonomously onboard manned and unmanned spacecraft in pressurized micro-gravity environments. In this paper we provide an overview of some of the design challenges we face in making the PSA practical, effective, and usable for future space missions. In particular we highlight the need for an agent architecture supporting adjustable autonomy and a generic model of teamwork.

Keywords

Agents, teamwork, adjustable autonomy, robotics

1. INTRODUCTION

Unmanned spacecraft must perform reliably and effectively in unpredictable precarious environments, despite strict limitations in their power, space, and computing resources. While in the past their activity has been planned and controlled by a relatively large and highly-skilled NASA mission operations team on the ground, this approach becomes less practical as missions of longer duration and distance become more common. Apart from any other consideration, the time it takes for a ground control instruction and a spacecraft response to make a round trip between the earth and deep space is simply too long. For these and other reasons, NASA has made research on onboard autonomy for unmanned spacecraft a high priority, to allow more rapid spacecraft response to potential problems and unforeseen science opportunities.

NASA is also interested in the promise of autonomous systems for manned missions. 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 1). 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.

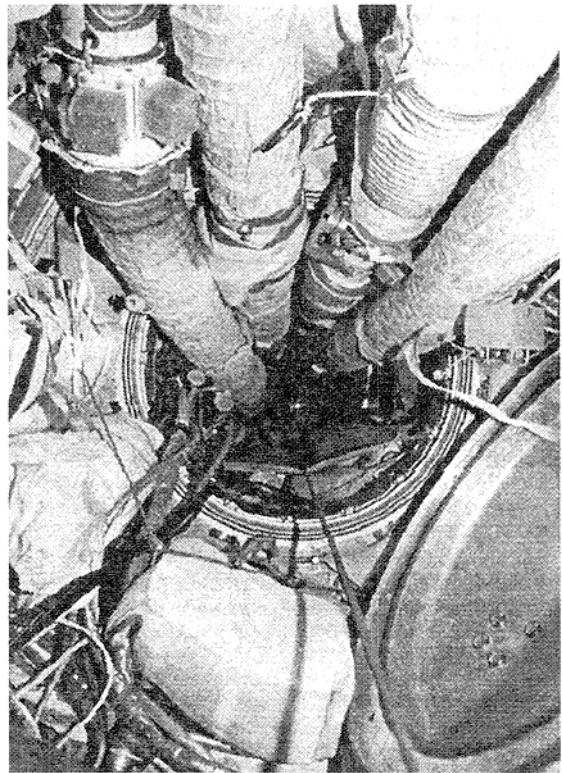
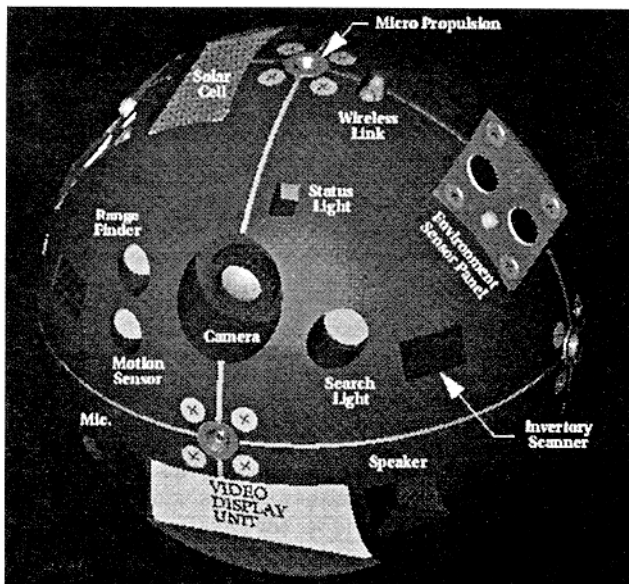


Figure 1. Obstructed hatchway between Mir modules

The Portable Satellite Assistant (PSA) is a softball-sized flying robot designed to operate autonomously onboard manned and unmanned spacecraft in micro-gravity, pressurized environments (figure 2). Environmental sensors for gas, temperature, and fire detection will provide the ability for the PSA to monitor spacecraft, payload and crew conditions. Video and audio interfaces will provide support for navigation, remote monitoring, and video-conferencing.



Ducted fans will provide propulsion and batteries will provide portable power.

2. MISSION SUPPORT SCENARIOS

The architecture of the PSA is designed to accommodate a wide range of components that enable a broad set of mission support scenarios:

- *Environmental health and status monitoring:* The PSA will patrol astronaut living quarters and laboratories and monitor oxygen, nitrogen, carbon dioxide, and other gas levels. The PSA will also monitor experimental data by visual inspection.
- *Communications:* The PSA will function as a remote, mobile data display terminal for system and experimental data. The PSA will have both communication input and output capabilities, including a microphone, speakers, camera, and flat display panel. These features allow for video and audio conferencing, as well as a wireless relay to link to hardwired communication systems.
- *Remote operations support:* Ground controllers or remote principal investigators will be able to communicate with the PSA and maneuver it into the desired location. Once in position, the PSA will, for example, use its onboard sensors to monitor the status of experimental lab animals in holding bays or provide the ground controllers with remote views of mission operations.
- *Crew worksite support:* The PSA provides astronauts work site access to such information as vehicle status and health data, mission schedule, inventory tracking, location information, and just-in-time-training support. The PSA also has the ability to provide notification updates and alarms related to system events, payload events, or mission experiments.

In later phases, we envision the PSA supporting fault isolation and recovery, with the ability to replace or augment sensor or controller capabilities at the point of need [19]. Inventory tracking could be performed autonomously. Additionally, integrated payload interfaces and cargo packages could be developed for injecting various supplies

such as food, experiment chemicals, and so forth into experimental units. For example, during a video inspection, a PSA could notice that specimens in habitat holding units needed food. One PSA would inject the supplies and another collaborating PSA would act as a supply cargo carrier.

3. PSA FOUNDATIONAL CAPABILITIES

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 [1]. 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 [9]. Optimal distribution of computing tasks among the various processors will be maintained by packaging code as mobile agents [7, 17].

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 [19].

Human interface. The PSA must support a variety of interfaces for the humans that interact with it [2]. 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.

4. AUTONOMY AND TEAMWORK

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*).

4.1 Adjustable Autonomy

The PSA's approach to autonomy draws on our experience in developing the Remote Agent for the Deep Space One mission [13, 15, 19] and more recent work in the area of Rover autonomy [8]. An intelligent executive will provide the dynamic planning for the PSA based on crew, payload, ground and spacecraft mission requirements.

Reasoning about the interaction between the various goals that an PSA could potentially achieve is a challenging and complex process. For operation at high levels of autonomy, the PSA must be able to take a high-level task specification and refine this request into a more detailed execution sequence. Once this sequence is constructed, however, the PSA must continue to adapt its actions to variations in the environment without requiring constant direction from some external source. For example, if the ground support team wants to look at a particular area of the station to perform a visual inspection, a high-level task request would be submitted to a PSA that describes the location that needs to be observed and the time by which the observation needs to occur. From this task request, it is the PSA's responsibility to select a path for reaching this destination while potentially performing other tasks.

This problem can be separated into two distinct tasks: 1) development of a high-level task language that can be used to command the PSA by describing a task that must be performed and providing information about how to accomplish this task when necessary, and 2) development of automated planning, scheduling and reactive execution techniques for reasoning about the tradeoffs between various tasks and for handling uncertainties within the environment in a reactive fashion. A key challenge is that unlike previous applications, most of the planning for this agent will need to be performed in a highly reactive manner with limited time for deliberative planning and scheduling to occur.

We have already developed an initial prototype of a model-based reactive execution language called RMPL that can be used to describe reactive control constructs. We are currently working on demonstrating the use of this language as a high-level specification language that can be used to describe tasks that must be performed by the PSA. Within the next two years, we expect to demonstrate the use of this language and explore how the PSA will reason about the tradeoffs between various simple task requests. This will be done both within a simulated environment as well as the hardware testbed.

Another 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 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 [11] while providing them a sufficient level of comfort that nothing will go wrong [14]. 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.

4.2 Teamwork

The PSA will need to perform many tasks involving cooperation with other PSA's, people, software agents participating in information access or performance support, data servers, and various electronic devices. While each of these heterogeneous cooperating entities operates at a different level of sophistication, they may each require some common means of representing and appropriately participating in joint goals.

One of the most promising approaches to maintaining coherence in dynamic teaming environments is based on an explicit general theory of teamwork (aka joint intention theory) [6, 18]. This approach is in contrast to the approach taken in most multi-agent systems where knowledge about maintaining team coherence, if it is explicitly represented at all, is modeled in an ad hoc domain-specific fashion.

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 teamwork model helps assure that each team member is notified under appropriate conditions of the failure without requiring special-purpose exception handling mechanisms to do this for each possible failure mode.

The PSA's agent-based teamwork capabilities will build on research in multi-agent communication, collaboration, and information access developed in KAoS as part of the NASA-sponsored Aviation Extranet project [3, 4, 10]. Teams will be formed, maintained, and disbanded through the process of agent-to-agent communication using an appropriate semantics [16]. 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 [5]. Agent *conversation policies* are being designed to assure robust

behavior and to keep computational overhead for team maintenance to an absolute minimum [12].

5. STATUS

Custom hardware components for the PSA have been fabricated including a custom air bearing assembly to float the PSA on an 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. We have developed a software simulation of the PSA using the Hybrid Concurrent Constraint (HCC) programming language in order to demonstrate goal-directed, reactive execution. Boeing's KAoS agent framework has been running at NASA Ames for several months and is being enhanced to support the PSA's more demanding requirements for teamwork, mobility, and fine-grained resource management. A joint effort with researchers from Stanford University to develop a characterization of the problem and an initial architecture for the environmental health monitoring task has been initiated.

6. CONCLUSIONS

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 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.

7. ACKNOWLEDGMENTS

We acknowledge the many contributions of the other members of the PSA project: Dan Clancy, Vineet Gupta, Beth Hockey, Frankie James, John Loch, and Mark Sibenac. We thank Mark Greaves and our anonymous reviewers for valuable comments.

8. REFERENCES

- [1] Alena, R. (1996), Wireless Network Experiment – Risk Mitigation Experiment 1306, report on STS-74/76-Mir20 experiment for the Phase One International Space Station Program.
- [2] Alena, R. (1997), Risk Mitigation Experiment 1329 – Space Station Crew Interface Functionality, report on STS-81 experiment for the Phase One International Space Station Program.
- [3] Bradshaw, J. M., Dufield, S., Benoit, P. & Woolley, J.D. (1997). KAoS: Toward an industrial-strength open agent architecture. In J.M. Bradshaw (Ed.) *Software Agents*. AAAI/MIT Press, pp. 375-418.
- [4] Bradshaw, J.M., Gawdiak, Y., Abou-Khalil, A., Carpenter, R., Cranfill, R., Jeffers, R., Kerstetter, M., Kulkarni, D., Poblete, L., Sun, A., Suri, N., (2000). Extranet applications of software agents. *ACM Interactions*, in preparation.
- [5] Bradshaw, J. M., Greaves, M., Holmback, H., Jansen, W., Karygiannis, T., Silverman, B., Suri, N., & Wong, A. (1999). Agents for the masses? In J. Hendler (Ed.) Special issue on agent technology, IEEE Intelligent Systems, March/April, 53-63.
- [6] Cohen, P. R. & Levesque, H.J. (1991). Teamwork. *Noûs*, 25 (4), 487-512.
- [7] Cybenko, G., Gray, R., Kotz, D. & Rus, D. (2000). Mobile agents: Motivations, state-of-the-art systems, and frontiers. In J. M. Bradshaw (Ed.) *Handbook of Agent Technology*, Cambridge, MA: AAAI/MIT Press, in press.
- [8] Dorais, G., Bonasso, R. P., Kortenkamp, D., Pell, B. & Schreckenghost, 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: AAAI Press.
- [9] Gawdiak, Y. (1993) Space Station Redesign Data Management System & Avionics Software - The implemented Option A/B portable architecture for the International Space Station
- [10] Gawdiak, Y. (1997). Aviation ExtraNet Joint Sponsored Research Plan. Approach for developing distributed extranet infrastructure to support Intelligent Agents.
- [11] Gershenfeld, N. (1999). *When Things Start to Think*. Cambridge, MA: MIT Press.
- [12] Greaves, M., Holmback, H., & Bradshaw, J. M. (1999). What is a conversation policy? In M. Greaves, and J. M. Bradshaw (Eds.), Proceedings of the Autonomous Agents '99 Workshop on Specifying and Implementing Conversation Policies, Seattle, Washington, May 1, 1-9.
- [13] Muscettola, N., Nayak, P., Pell, B. & Williams, B. C. (1998). Remote Agent: To Boldly Go Where No AI System Has Gone Before. *Artificial Intelligence* 103 (1-2):5-48, August.
- [14] Norman, D. A. (1997). How might people interact with agents? In J.M. Bradshaw (Ed.) *Software Agents*. AAAI/MIT Press, pp. 49-55.
- [15] Pell, B., Bernard, D. E., Chien, S. A., Gat, E., Muscettola, N., Nayak, P., Wagner, M.D., & Williams, B. C. (1997) An autonomous spacecraft agent prototype. In Proceedings of the First International Conference on Autonomous Agents, Marina del Rey, CA
- [16] Smith, I.A., Cohen, P.R., Bradshaw, J.M., Greaves, M. & Holmback, H. (1998). Designing conversation policies using joint intention theory. Proceedings of the International Joint Conference on Multi-Agent Systems (ICMAS-98), Paris, France, 2-8 July 1998, Los Alamitos, CA: IEEE Computer Society, 269-276.
- [17] Suri, N., Bradshaw, J. M., Breedy, M. R., Groth, P. T., Hill, G. A., Jeffers, R., Mitrovich, T. S., Pouliot, B. R., Smith, D. S. (1999). NOMADS: Toward an environment for strong and safe mobility. Submitted for publication.
- [18] Tambe, M., Shen, W., Mataric, M., Pynadath, D. V., Goldberg, D., Modi, P. J., Qiu, Z. & Salemi, B. Teamwork in cyberspace: Using TEAMCORE to make agents team-ready. Proceedings of the AAAI Spring Symposium on Agents in Cyberspace. Menlo Park, CA: AAAI Press.
- [19] Williams, B. & Gupta, V. Unifying model-based and reactive programming within a model-based executive. In Principles of Diagnosis Workshop 1999.