NASA's Mobile Agents Architecture: A Multi-Agent Workflow and Communication System for Planetary Exploration

Maarten Sierhuis[1, 2], William J. Clancey[2], Richard L. Alena[2], Dan Berrios[3, 2], Simon Buckingham Shum[5], John Dowding[3, 2], Jeff Graham[6], Ron van Hoof[3, 2], Charis Kaskiris[7], Shannon Rupert[8], Kim S. Tyree[9]

[1]USRA/RIACS, [3]UARC, [4]QSS, Inc.,
[2]NASA/Ames Research Center, Moffett Field, CA, USA
{msierhuis, berrios, jdowding, rvhoof}@mail.arc.nasa.gov
{William.J.Clancey, Richard.L.Alena}@nasa.gov
[5]Knowledge Media Institute, Open University, Milton Keynes, UK
S.Buckingham.Shum@open.ac.uk
[6] S&K Technologies, Inc., NASA/Johnson Space Center, Houston, TX, USA
jgraham@sk-tech.com
[7]School for Information Management and Systems, University of California at Berkeley, CA, USA
kaskiris@sims.berkeley.edu
[8]The Mars Society, Mira Costs College, CA USA
srupert@miracosta.edu
[9]NASA/Johnson Space Center, Houston, TX, USA
Kim.S.Tyree@nasa.gov

Abstract

NASA Ames' Mobile Agents Architecture is a distributed agent-based architecture, which integrates diverse mobile entities in a wide-area wireless system for lunar and planetary surface operations. Software agents, implemented in the Brahms multiagent language, run in Brahms virtual machines onboard laptops for space suits, robots, and surface habitats. "Personal agents" support the habitat crew and surface astronauts, as well as the their robotic assistant. People communicate with their personal agents via a speech dialogue system and via a meeting-capture hyperlink database tool.

1 INTRODUCTION

NASA Ames' Mobile Agents Architecture (MAA) is a distributed agent-based architecture, which integrates diverse mobile entities in a wide-area wireless system for lunar and planetary surface operations. The architecture was first developed in 2001, and has been extensively tested in two-week field campaigns at NASA Johnson Space Center in 2001, at the Meteor-Crater near Flagstaff, AZ, in 2002, and the Mars Society's Desert Research Station (MDRS) during April 2003[1] and 2004[2]. Software agents, implemented in the Brahms multiagent language, run in Brahms virtual machines onboard laptops for space suits, robots, and surface habitats. "Personal agents" support the habitat crew and surface astronauts, as well as the their robotic assistant (a.k.a. proxy agents in the MAS community, however personal agents do not represent the user, like avatars, but rather are assistants to the user).

People communicate with their personal agents via their speech dialogue system and via a discussion-capture hyperlink database tool. The personal agents coordinate the surface exploration work. The agents process GPS, health data, and voice commands—monitoring, controlling and logging science data throughout simulated EVAs with two geologists and a robot.

Communications are maintained over wireless nodes distributed over hills and into canyons; agents automatically transmit and store science data to a semantic web database in the surface habitat and then via satellite to the "Earth-based" semantic web database. An e-mail agent running in the surface habitat sends alerts to remote scientists and mission support spread out all over the world.

[1] http://www.marssociety.org/MDRS/fs02/crew16/
[2] http://www.marssociety.org/MDRS/fs03/crew29/

Proc. of 'The 8th International Symposium on Artifical Intelligence, Robotics and Automation in Space - iSAIRAS', Munich, Germany. 5-8 September 2005, (ESA SP-603, August 2005)

Over the past four years, the MAA has become a robust architecture that has been tested in actual field exploration conditions in the Utah desert, in a wide-area network setup spanning more than 5 km from the MDRS habitat, supporting two EVA astronauts and an EVA Robotic Assistant (ERA) in geological surface science exploration, communicating with the crew in the habitat and remote scientists all over the world via a satellite connection to the internet [10]. Special care has been taken to provide robust communications between agents in case communication falls away in this fragile planetary wide-area wireless network. Over the years, latency in the architecture, due to agent belief-set increases, network latency and its impact on agent reasoning, have been addressed.

One role of the personal agents in the MAA is to coordinate communication between and activities for people and robots. Another role of the personal agents is to enable the capture, correlation, transmission and storage of all sorts of science data, as well as low-level architecture and personal telemetry data. A third role is to monitor an astronaut's EVA plan execution providing warnings in case the astronauts veer of the plan, as well as performing autonomous EVA plan execution for the ERA robot. The role of the astronaut's personal agents is also to monitor the astronaut's health using a number of biosensors, providing initial healthcare monitoring to the astronauts, as well as monitoring battery levels and temperature of devices in the architecture. The personal agents also maintain second-by-second location information of the astronaut and robot, using a 1.5 cm accurate differential GPS service provided to each personal agent by the ERA robot.

This paper describes the Mobile Agents Architecture from the 2004 field campaign.

2 FIELD CAMPAIGNS

This section provides an overview of the Mobile Agents Project field campaigns. These campaigns serve two purposes: as a testbed for testing the MAA, and more important, as a way to gather future work system requirements, based on a study of the use of the system in a realistic environment (rather than a laboratory context that always has a simplification from reality). This is essential to a human-centered design approach.

2.1 Mars Desert Research Station

Every April the Mobile Agents team goes to the realistic Mars analog site of the Mars Society, the Mars Desert Research Station (MDRS) in the Utah Desert, and tests the next version of the Mobile Agent Architecture with one or two EVA scenarios.

The MDRS is a simulated Mars habitat designed and built by the American Mars Society[3]. Figure 1 shows the MDRS located in the Utah desert near Hanksville, in the continental USA.

A crew of six lives inside the habitat and pretends they are living and working on Mars performing geological science extra-vehicular activities (EVA).

2.2 EVA Test Scenario

In April 2004, the team tested the following scenario:

Imagine an astronaut crew on Mars planning to explore a, distant from the habitat, geologically interesting site. Their normal procedure for doing this is in a two-day EVA, using a robot the first day to go on an autonomous EVA to the site to do site-reconnaissance.

Figure 1. The MDRS near Hanksville, Utah,

The crew is in the habitat monitoring the robot's EVA and viewing the images (both streaming video and panorama images) being send back to the habitat. Using the images captured by the robot, the crew starts planning their site exploration EVA where two EVA astronauts, supported by the robot, will go to the site and do a geological survey (i.e. taking more detailed images of interesting sites and bringing back rock and soil samples).

Just as the crew has been looking at the robot's returned images, a team of geographically distributed remote scientists, back on Earth (the RST), has received e-mail notifications of the robot's activities and science data. The e-mails contain URLs to the images stored in a semantic-web database accessible by the Earth-based team. Each RST member is individually monitoring and analyzing the data, depending on the time of day in their local time zones.

When the crew is finished planning their next day's EVA, they download their EVA plan to the RST members, and call it a long-work day on Mars. While the crew is sleeping, the RST members on Earth are discussing the crew's EVA plan for the next day. The RST sends their comments on the plan back to the crew, before the crew wakes up. When the crew wakes up they look at the RST's comments and finalize their EVA plan, and then set out on their day's work outside the habitat. They don their space suits, start the robot, and load their EVA plans into their personal agents.

The robot follows the EVA astronauts on their way to the exploration site. When they arrive, the EVA astronauts go to

[3] http://www.marssociety.org

work, taking pictures, digital voice notes, and samples—similar to the Apollo astronauts on the Moon more than 30 years ago, accept that roles that were played by people in mission control at Johnson Space Center, are now played by software agents. They are asking the robot to move to specific locations to take photographs and panoramas and ask it to follow them, so that they can move around. The robot's role is to support them in their exploration activity, carrying their tools and soil and rock samples and being a video- and photographer. An important function of the robot is that of a mobile wireless relay station. During the EVA, all telemetry and science data is automatically captured, correlated and send back to the habitat, where it is stored and communicated back to the RST members on Earth. When the astronauts come back in the habitat they continue their exploration work by analyzing the samples, taking more close-up photos of the samples and notes about them. All this information is also stored and communicated back to the RST.

3 THE MOBILE AGENTS ARCHITECTURE

In this section we describe the Mobile Agent Architecture as it was implemented and tested for the April 2004 field test. We are currently redesigning parts of the architecture and adding new capabilities, based on our experiences from the field test. Here we only discuss the architecture as it pertains to the 2004 field test.

3.1 Hardware and software entities

The current version of the MAA supports the Mars crew, and to some extent the RST, in performing the above scenario.

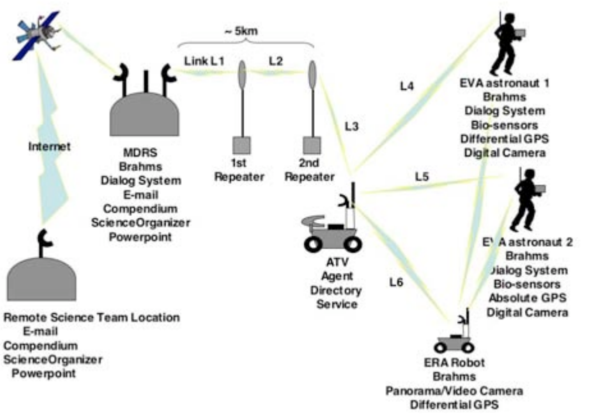

Figure 2. Mobile Agents entities and wireless network

Figure 2 depicts all the (mobile) hardware entities and their software components, as well as how they are connected via the Mobile EXploration (MEX) wireless network [2]. Starting on the right side of Figure 2 we have the two EVA astronauts who are wearing a mockup-space suit with backpack (see Figure 3). The backpack includes the wireless network components, biosensors, GPS equipment and a laptop computer running all the software components, including the astronaut's speech dialogue system and Brahms.

Figure 3. EVA Astronaut's space suit and backpack

Bellow the EVA astronauts, in Figure 2, there is Johnson Space Center's EVA Robotic Assistant (ERA Robot) [24]. Besides computers running the low-level robotic architecture, including video and panorama cameras, and the navigation and obstacle avoidance systems, the robot also has a laptop computer running agents in a Brahms VM (see Figure 4).

Figure 4. EVA Robotic Assistant

Next to the ERA, in Figure 2, is the MEX all-terrain-vehicle (ATV) (see Figure 5). The MEX ATV is the central wireless network hub between the "mobile agents" and the habitat. The MEX ATV runs the KAoS agent directory server [3] enabling the Brahms agents to find each other on the network.

Next, in Figure 2, is the Mobile Agents software configuration in the habitat. There is a laptop computer in the habitat running Brahms and a dialogue system for the HabCom crewmember (see bellow for an explanation of the HabCom role), as well as the ScienceOrganizer semantic-web database [13] and the Compendium software tool. Compendium is essentially a graphical hypertext system that allows a group to incrementally build a collaborative mind map of a discussion [17].

Figure 5. MEX All-Terrain-Vehicle

The last entity in Figure 2 is the RST. Although the RST is depicted in Figure 2 as being in one location, in reality the RST is a distributed team in six different locations with three different time zones, in the US and UK. The RST members have both access to ScienceOrganizer and Compendium, as well as e-mail and WebEx™. However, there is currently no agent support for the RST.

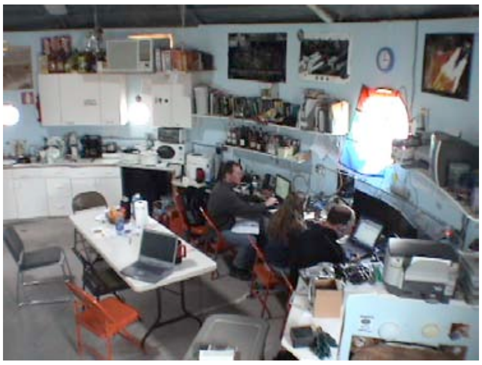

Figure 6. HabCom in the MDRS habitat

3.2 Brahms

The MAA is a distributed agent architecture developed in Brahms. Brahms is a multiagent rule-based BDI language developed at NYNEX S&T, IRL and NASA Ames. The Brahms environment consists of a language definition and compiler, a graphical development environment (the Composer) and a Brahms virtual machine (the BVM), running on top of the Java virtual machine, to load and execute Brahms agents. Brahms was originally developed as a multiagent language for modeling and simulating human work practice behavior in organizations [9]. While Brahms can run in simulation mode, and is still used as a simulation environment [21], we have extended the BVM by allowing agents to run as real-time software agents without a simulation clock and event scheduler to synchronize the agents. This makes Brahms both a simulation and a software agent development environment. With Brahms you can test a multiagent system by running the system as a simulation. When the system is debugged (using the Brahms AgentViewer), you can "flip the switch" and run the same system as a real-time distributed agent system. We refer to this as "from simulation to implementation," a software engineering method that uses simulation as a system design and integration test environment.

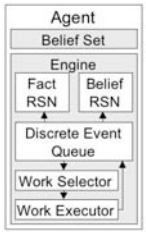

Figure 7. Brahms agent engine

Brahms agents are rule-based BDI-like agents. However, Brahms does not use a goal-directed approach, but rather an approach we refer to as activity-based. Brahms agents are both deliberative and reactive. Each Brahms agent has a separate subsumption-based inference engine (see Figure 7) [5]. Brahms agents execute multiple activities at different levels at the same time. At each belief-event change (creation or changing of beliefs), situated-action rules (i.e. workframes) (see Figure 8) and production rules (thoughtframes), at every active activity-level, are evaluated (see Figure 9).

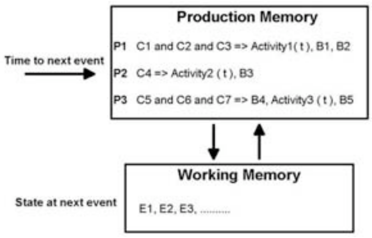

Figure 8. Situated-action rules, a.k.a. workframes

Based on activity priorities, the engine determines which activity in the workframe-activity subsumption hierarchy should be the current active activity. This way, agents can switch activity context at any moment, interrupting one activity to start another and returning to where an interrupted activity left off, when no other higher priority activity is active.

Agents can change their beliefs by reasoning, detection of facts in the world model, or by communicating with other agents. Agents can communicate beliefs to each other by using a special type of activity, called a communication activity.

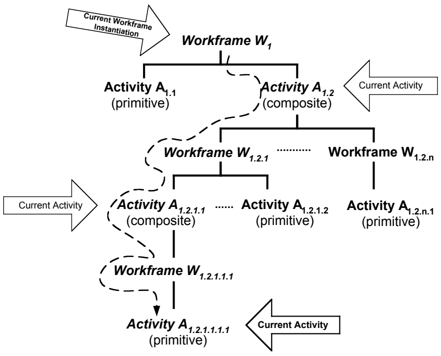

Figure 9. Workframe-Activity Subsumption Hierarchy

Brahms is a modeling language designed to model human activity. Agents, therefore, were developed to represent people. Brahms agents can belong to one or more group, inheriting attributes, initial beliefs, and activities, workframes and thoughtframes from multiple groups (multiple inheritance). This allows the abstraction of agent behavior into one or more groups. Because Brahms was developed to represent people's activities in real-world context, Brahms also allows the representation of artifacts, data and concepts in the form of classes and objects. Both agents and objects can be located in a model of the world—the geography model—giving agents the ability to detect objects and other agents in the world and have beliefs about objects. Agents can move from one location in the world to another by executing a move activity, simulating the movement of people. For a more detailed description of the Brahms language we refer the reader to [18], [20] and [1].

An important aspect of making Brahms into a software agent development language is the ability to seamlessly integrate Brahms agents within Java. Brahms has a JAPI defined to write agent activities in Java, so that they can be called from workframes. Brahms agents can also be completely written in Java, which enables the wrapping of existing external systems as a Brahms agent, enabling Brahms agents to communicate beliefs to and from external systems (see Figure 10). So-called external (Java) agents are frequently used in the MAA to communicate with biosensors, GPS devices, speech dialogue system, databases, et cetera.

3.3 Distributed Brahms

For the MAA, the Brahms environment needed agents to be distributed over a network of computers, each computer running a number of agents communicating with agents running on other computers over the network. Each Brahms virtual machine, running a Brahms model with a number of agents, is local to a computer, similar to a Java VM running multiple Java threads (actually, each Brahms agent in a BVM runs as a single Java thread).

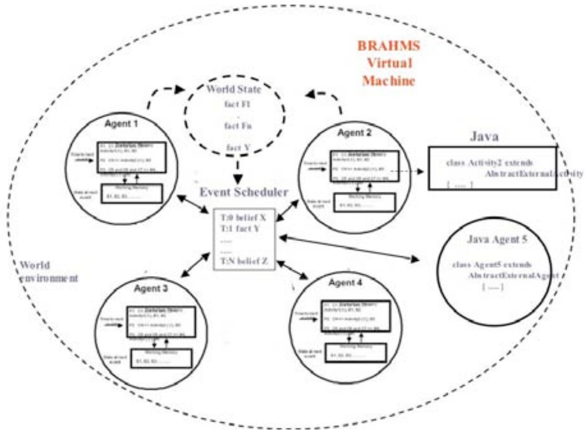

Figure 10. Brahms VM with Java activities and agents

To enable multiple BVM's to connect to each other in a distributed fashion, allowing the agents within each BVM to find and communicate with each other, we integrated Brahms with the KAoS agent communication framework [4]. Each BVM contains a Brahms KAoS Service registering every Brahms agent and object with the KAoS Domain Manager at startup (see Figure 11). Every Brahms agent outside of a BVM that is being referenced by an agent from within that BVM is found in the KAoS domain server. The KAoS transport layer (Corba) allows any Brahms agent to communicate with any other Brahms agent in its domain, as if it is running in its local BVM.

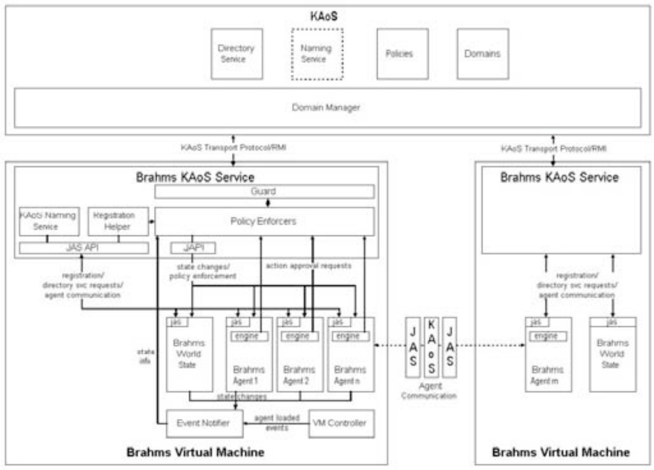

Figure 11. Distributed Brahms/KAoS Architecture

3.4 The Agent Architecture

Figure 12 shows the Brahms agents of and the external systems integrated with the MAA. There are three types of mobile entities that are currently being supported, the EVA astronaut, the ERA robot and the HabCom crewmember in the habitat.

Each entity has a support computer with a Brahms model running within a local Brahms VM: The Space Suit Brahms

VM (one for each astronaut), the ERA Brahms VM and the HabCom Brahms VM. Each Brahms model has a number of agents that support the mobile entity. As discussed above, each agent can communicate with remote agents via KAoS through the MEX wireless network.

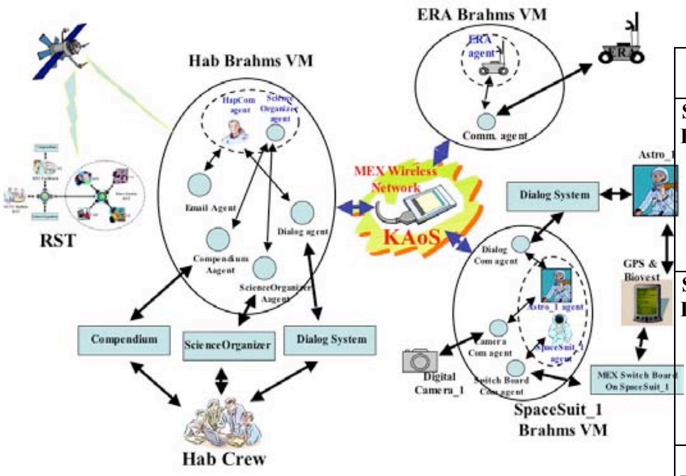

Figure 12. The agents in the Mobile Agents Architecture

Each Brahms model has three types of agents, Personal Agents (PA), Task Agents (TA) and Communication Agents (CA). PA's assist an autonomous external entity (person, robot) in performing their activities during an EVA. TA's assist PA's in executing low-level tasks associated with a particular activity during the EVA. CA's agentify external systems, so that the PA's and/or TA's can communicate with an external system as if communicating with another agent. CA's translate MA speech acts into a dedicate message format for the external system and vice versa (see agent communication section bellow). Table 1 shows the functions each agent type supports. Not every agent supports all functions listed, and no VM has all the same agents running in it.

Table 1. MAA Agent Functions

Agent Type	Supported Functions
Personal Agent	Track Location; Monitor Biosensors; System Alerting and Notification; Take Picture; Monitor Batteries; Take Panorama; Take Digital Voice Notes; Create Sample Bag; Curate Sample Bag; Name Image; FTP Image and Voice Note; Create New Location; Associate Image, Voice Note, Location, Plan Activity, EVA Plan; EVA Plan Distribution; Start EVA Plan Activity; Read EVA Plan; Monitor EVA Plan; Command Robot to Move, Follow, Take Picture, Take Panorama; E-mail Notification;
Task Agent	Images Download; GPS Tracking; Biosensor Monitoring; Science Data Management; Autonomous ERA Plan Execution;
Communication Agent	Dialog System Communication; ScienceOrganizer Communication; Compendium Communication; Camera Communication; Sensor Data Communication (GPS + Biosensor); ERA Robot Communication;

Table 2 provides the main agents running in each Brahms VM shown in Figure 12.

Table 2. MAA Agent Distribution

	Personal Agents	Task Agents	Communication Agents
SpaceSuit_1 Brahms VM	AstroOnePA	AstroOneCamera TA SpaceSuitOneTA LocationManager AstroOneTimeSta mpKeeperCA	AstroOneDA AstroOneCamera CA SpaceSuitOneMe xCA SpaceSuitOneBatt eryCA
SpaceSuit_2 Brahms VM	AstroTwoPA	AstrotwoCamera TA SpaceSuitTwoTA AstroTwoTimeSt ampKeeperCA	AstroTwoDA AstroTwoCamera CA SpaceSuitTwoMe xCA SpaceSuitTwoBat teryCA
ERA Brahms VM	EraPA	EraPlanExecutor EraCameraTA	EraCA
Hab Brahms VM	HabComPA	ScienceOrganizer TA	HabComDA ScienceOrganizer CA CompendiumCA

4 AGENT COMMUNICATION

In this section we briefly describe how agents, robots, systems and people communicate information to each other in the MAA.

4.1 Speech Acts

The way agents communicate with each other in the MAA is via speech act objects. A *SpeechActObject* specifies an utterance of one agent to another and is used by agents to communicate with one another. SpeechActObjects are compliant with the FIPA Communicative Act Library [12]. Currently the MAA supports the message types: inform, request, accept, and failure. For our 2005 field campaign we are implementing the subscribe type.

4.2 Human-Agent Dialogue

The RIALIST Dialog System in the MAA is the system used to allow the crew (HabCom and EVA astronauts) to interact with their personal agents using speech [11]. It is responsible for interpreting spoken commands from the user and translating it into SpeechActObjects, as well as interpreting SpeechActObjects into speech.

In order for the Dialog System to interpret speech to Brahms SpeechActObjects and agent beliefs, and vice versa, a dialog communication agent (DialogCA) agent is required. Figure 13 gives an overview of the systems that are part of the RIALIST Dialog System and shows how Brahms is connected to the Dialog System's OAA architecture [7].

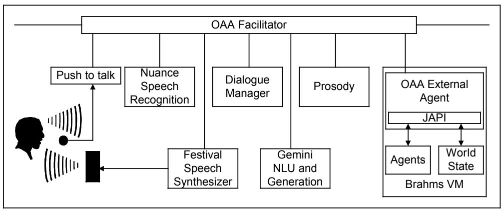

Figure 13. RIALIST Dialog System with DialogCA (OAA External Agent)

Examples of supported speech commands are listed in Table 3:

Table 3. Example Speech Commands

Human Speech Request	Recipient
"Boudreaux take a [panorama | picture]"	EraPA
"Boudreaux [follow | watch] me"	EraPA
"Boudreaux move to {location X}"	EraPA
"Start tracking my [GPS location | biosensors] every 10 seconds"	AstroOnePA or AstroTwoPA
"What is my current activity?"	AstroOnePA or AstroTwoPA
"How much time do I have left?"	AstroOnePA or AstroTwoPA
"Download all images"	AstroOnePA or AstroTwoPA
"Label last image {X}"	AstroOnePA or AstroTwoPA
"Create a sample bag and call it {X}"	AstroOnePA or AstroTwoPA
"Associate sample bag called {X} with the last voice note"	AstroOnePA or AstroTwoPA

4.3 Agent-Robot Communication

In order for the Brahms EraPA to communicate with the ERA robot to control its actions and to get feedback, there is an ERA communication agent (EraCA). The ERA provides a CORBA interface to the ERA's actions. The EraCA is responsible for translating any speech act requests to the EraPA into the appropriate CORBA requests and is also responsible for translating any results returned by the ERA through callbacks into the appropriate SpeechActObject reply (see Figure 14).

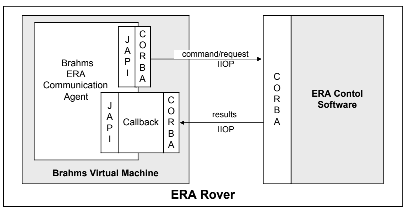

Figure 14. Era Communication Agent

5 RELATED WORK

The use of proxy-type agents as a way to coordinate activities between people and agents has been researched by others, e.g. [23], [15]. Most of these architectures run as simulations or small experimental systems in laboratory settings, and explore the different styles of coordination and the effect on team performance.

Research with "human and robot in the loop" decision-making in multiagent systems has not yet been studied much. Scerri, et al aim to gain insight to the benefits of giving people the opportunity to coordination decisions in teams of robots, agents and persons (RAP) [16]. Murphy's work on the coordination of urban search and rescue (USAR) robots by the human operator focuses on determining the desired characteristics of USAR robots, so that the number of humans in the human-robot team can be reduced [14]. Murphy's work is grounded in applying robotics in actual emergency situations, and applies the notion of human-centered design from her experience in real-life situations.

The new field of human-robot interaction (HRI) is starting to focus its research more broadly on not only one-to-one interaction between the human operator and the robot, but also on the context of the complete work system [6] [8].

6 DISCUSSION

In this paper we presented the Mobile Agents Architecture, a proxy-agent architecture for supporting exploration of planetary surfaces. We take a human-centered design approach where the requirements for the architecture's agent capabilities are derived from repeated experimentation in real-world analog environments. In short, we let the requirements be derived from the work practice of the people, creating an agent technology research pull instead of a push. We discussed how our human-centered design approach, based on experimentation in a Mars analog setting, is instrumental to our MAA research.

The MAA consists of distributed Brahms agents running in Brahms VM's in a wide-area wireless network. Each Brahms VM has three types of agents, personal-, task- and communication agents, which together support a person or robot.

One of the practical benefits of the MAA is the ease with which we can add new or existing external systems to the architecture, without impacting existing pieces. This enables us to quickly improve or change the capability of the complete system. For example, after our April 2004 field test we added a PER[4] robot to the MAA. To do this we created another member of the RobotAgent group, called the PerPA, and added a PerCA agent that interfaced to the PER robot's architecture. Without change to the rest of the architecture we are now able to use both robots. Another experienced benefit is the way external systems can be developed independent of each other, and by integrating them in the MAA they are able to communicate and work together. For example, the ScienceOrganizer team never needed to interact with the robot team, but after both were independently integrated in the MAA the robot was able to store its images and panoramas in the ScienceOrganizer semantic database.

---

[4] http://www-2.cs.cmu.edu/~personalrover/PER/

A current limitation of the architecture is the centralized EVA plan monitoring done by the HabComPA. When communication to the habitat falls away, the PA's cannot monitor the astronauts and robot's EVA plan. This year we are distributing individual PA plan monitoring, execution and coordination. Another limitation is the lack of a teamwork model for the PA's. Imagine both astronauts want to use the ERA robot, who shall it serve and how does it make that decision? Currently, the EraPA has a FIFO queue for command handling. We are working on implementing our teamwork model [19]. A third limitation is a problem with maintaining network communication. Although the PA's have network failure handling capabilities, for safety reasons the astronauts should always be in network communication with the habitat. Knowing when network will fall away and autonomously moving to keep the network intact, is an important role for the robot (the robot as an intelligent mobile antenna). Also, at this moment we have to predefine GPS locations for the EVA plan. This means we have to pre-scout an area for "waypoints", before we create an EVA plan. We are working on enabling the ERA to autonomously scout an area and a) create waypoints and b) create a network signal-strength map of the area. Another major weakness is the lack of long-term memory for the agents. This a) limits the duration an agent can run, before the increased belief-set slows down agent reasoning (currently we can run for more than eight hours, but the objective is to run for weeks on end) and b) limits the agent's use of experiential knowledge from previous EVA's [22]. The current MAA is not yet a complete "plug n' play" agent architecture. Currently, agents need to be started together at system initialization time. Agents should be able to plug in and the system finds a "driver" and registers to the rest of the agents. This includes adding peripherals, e.g., a robot arm, which would be known generically and managed by existing agents (that might be only located on one platform and dynamically uploaded and "linked" into a particular model. We are working on improving many of these limitations for our next field campaign in April 2005.

Acknowledgements

This research was sponsored by the National Aeronautics and Space Administration's Computing, Information and Communications Technology Program, as part of human-centered systems area in the intelligent Systems Project. The authors would like to thank all those who participate in the Mobile Agents project.

7 REFERENCES

1. AiS. Brahms Website: http://www.agentisolutions.com, Ron van Hoof, 2000.
2. Alena, R.L., Ossenfort, J., Lee, C., Walker, E. and Stone, T., Design of Hybrid Mobile Communication Networks for Planetary Exploration. in *IEEE Aerospace Conference*, (Big Sky, MT, 2004).
3. Bradshaw, J.M., Dutfield, S., Benoit, P. and Woolley, J.D. KAoS: Toward an industrial-strength generic agent architecture. in Bradshaw, J.M. ed. *Software Agents*, AAAI Press/The MIT Press, Cambridge, MA, 1997, 375-418.
4. Bradshaw, J.M., M.Sierhuis, Acquist, A., Feltovich, P., Hoffman, R., Jeffers, R., Prescott, D., Suri, N., Uszok, A. and Hoof, R.v. Adjustable Autonomy and Human-Agent Teamwork in Practice: An Interim Report on Space Applications. in Hexmoor, H., Castelfranchi, C. and Flacone, R. eds. *Agent Autonomy*, Kluwer, 2003.
5. Brooks, R., A. Intelligence without representation. *Artificial Intelligence*, *47*. 139-159.
6. Burke, J., Murphy, R.R., Rogers, E., Scholtz, J. and Lumelsky, V. Final Report for the DARPA/NSF Interdisciplinary Study on Human-Robot Interaction. *IEEE Systems, Man and Cybernetics Part A*, *34* (2).
7. Cheyer, A. and David, M. he Open Agent Architecture. *Journal of Autonomous Agents and Multi-agent Systems, special issue on "Best of ICMAS 98"*, *4* (1). 143-148.
8. Clancey, W.J. Roles for Agent Assistants in Field Science: Understanding Personal Projects and Collaboration. *IEEE Transactions on Systems, Man and Cybernetics—Part C: Applications and Reviews*, *34* (2).
9. Clancey, W.J., Sachs, P., Sierhuis, M. and van Hoof, R. Brahms: Simulating practice for work systems design. *International Journal on Human-Computer Studies*, *49*. 831-865.
10. Clancey, W.J., Sierhuis, M., Alena, R., Crowford, S., Dowding, J., Graham, J., Kaskiris, C., Tyree, K.S. and Hoof, R.v., The Mobile Agents Integrated Field Test: Mars Dessert Research Station 2003. in *FLAIRS 2004*, (Miami Beach, Florida, 2004).
11. Dowding, J. and Campana, E., Spoken Dialogue Systems in MDRS Rotation 29. in *7th International Mars Society Convention*, (Chicago, IL, 2004).
12. FIPA. FIPA Communicative Act Library Specification, Foundation for Intelligent Physical Agents, 2002.
13. Keller, R.M., Berrios, D.C., Carvalho, R.E., Hall, D.R., Rich, S.J., Sturken, I.B., Swanson, K.J. and Wolfe, S.R., SemanticOrganizer: A Customizable Semantic Repository for Distributed NASA Project Teams. in *ISWC2004, Third Intnl. Semantic Web Conference*, (Hiroshima, Japan, 2004), Springer-Verlag, Berlin.
14. Murphy, R. Human Robot Interaction in Rescue Robotics. *IEEE Systems, Man and Cybernetics Part C: Applications and Reviews, special issue on Human-Robot Interaction*, *34* (2).
15. Pynadath, D.N. and Tambe, M. An Automated Teamwork Infrastructure for Heterogeneous Software Agents and Humans. *Autonomous Agents and Multi-Agent Systems*, *7*. 71-100.
16. Scerri, P., Pynadath, D., Johnson, L., Rosenbloom, P. and Si, M., A Prototype Infrastructure for Distributed Robot-Agent-Person Teams. in *2nd International Joint Conference on Autonomous Agents and Multiagent Systems*, (Melbourne, Australia, 2003), ACM Press, 433-440.
17. Selvin, A.M. and Shum, S.J.B. Rapid Knowledge Construction: A Case Study in Corporate Contingency Planning Using Collaborative Hypermedia. *Knowledge and Process Management*, *9* (2). 119-128.
18. Sierhuis, M. *Modeling and Simulating Work Practice; Brahms: A multiagent modeling and simulation language for work system analysis and design*. University of Amsterdam,

SIKS Dissertation Series No. 2001-10, Amsterdam, The Netherlands, 2001.

19. Sierhuis, M., Bradshaw, J.M., Acquisti, A., Hoof, R.v., Jeffers, R. and Uszok, A., Human-Agent Teamwork and Adjustable Autonomy in Practice. in *The 7th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS)*, (Nara, Japan, 2003).
20. Sierhuis, M., Clancey, W.J. and Hoof, R.v. Brahms: A multiagent modeling environment for simulating work practice in organizations. *Journal for Simulation Modelling Practice and Theory, Elsevier, The Netherlands*, *Special issue on Simulating Organisational Processes*.
21. Sierhuis, M., Clancey, W.J., Seah, C., Trimble, J.P. and Sims, M.H. Modeling and Simulation for Mission Operations Work System Design. *Journal of Management Information Systems*, *Vol. 19* (No. 4). 85-129.
22. Sierhuis, M. and Hoof, R.v., On Short-Term and Long-Term Memory for Brahms Agents. in *AAAI Spring Symposium 2004 Workshop on Interaction between Humans and Autonomous Systems over Extended Operations*, (Stanford University, CA, 2004), AAAI Press, 29-34.
23. Tambe, M. Towards Flexible Teamwork. *Journal of Artificial Intelligence Research*, *7*. 83-124.
24. Tyree, K.S., Robotic Assistance for Human Planetary and Lunar Exploration. in *AIAA Space 2004 Conference*, (San Diego, CA., 2004).