Brahms Mobile Agents: Architecture and Field Tests

William J. Clancey, Maarten Sierhuis, Charis Kaskiris, and Ron van Hoof

NASA/Ames Research Center
Computational Science Division, MS 269-3
Moffett Field, California 943055
bclancey@mail.arc.nasa.gov

Abstract

We have developed a model-based, distributed architecture that integrates diverse components in a system designed for lunar and planetary surface operations: an astronaut's space suit, cameras, rover/All-Terrain Vehicle (ATV), robotic assistant, other personnel in a local habitat, and a remote mission support team (with time delay). Software processes, called "agents," implemented in the Brahms language (Clancey, et al. 1998; Sierhuis 2001), run on multiple, mobile platforms. These "mobile agents" interpret and transform available data to help people and robotic systems coordinate their actions to make operations more safe and efficient. 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. A state-of-the-art spoken dialogue interface is integrated with Brahms models, supporting a speech-driven field observation record and rover command system (e.g., "return here later and bring this back to the habitat"). This combination of agents, rover, and model-based spoken dialogue interface constitutes a "personal assistant." An important aspect of the methodology involves first simulating the entire system in Brahms, then configuring the agents into a run-time system

Team Members

This project is a collaboration across NASA centers and other organizations:

- Brahms Project Group (NASA/ARC: W.J. Clancey, Principal Investigator; M. Sierhuis, Project Manager; R. van Hoof, lead programmer; C. Kaskiris, modeler)
- RIALIST Voice Commanding Group (RIACS: J. Dowding, Jim Hieronymous)
- MEX Vehicle & Wireless Communications Group (ARC: R. Alena)
- Exploration Robotic Assistant Group (NASA/JSC: D. Kortenkamp, R. Burridge, J. Graham)
- Space Suit Design (Stanford: S. Crawford, in collaboration with J. Kosmo, JSC).

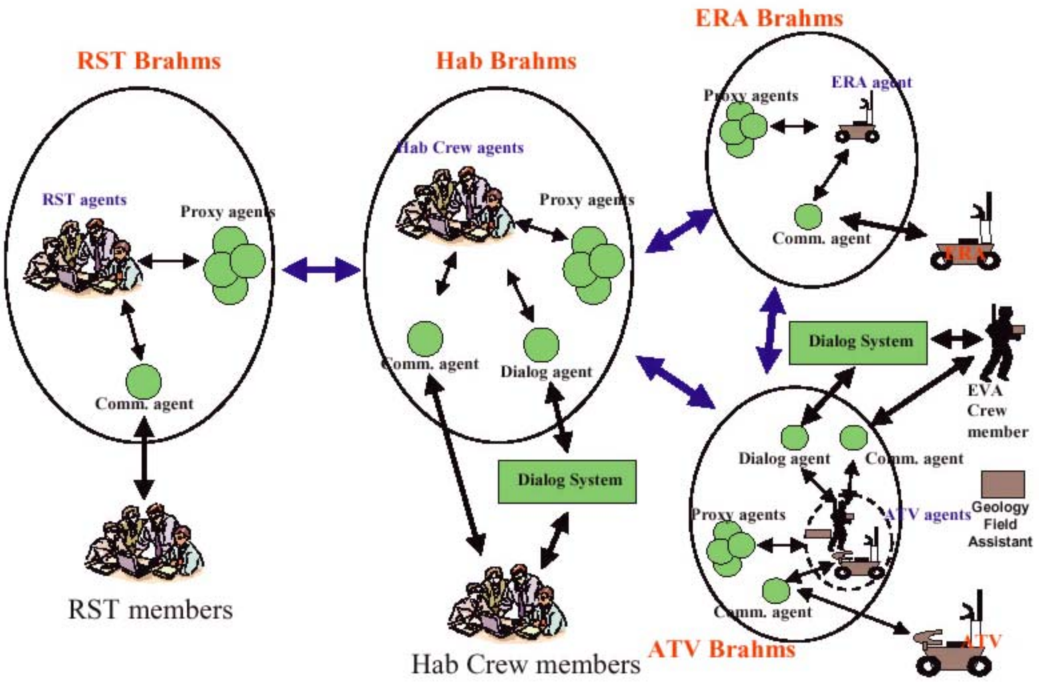

Figure 1. Brahms Mobile Agent Architecture (MAA)

Scenario Architecture

In this generalized scenario, a Remote Science Team (RST) oversees and advises surface operations that are locally managed by crew members in the habitat. On the surface, participating in an Extra-Vehicular Activity (EVA), are the Exploration Robotic Assistant (ERA) and crew members using ATVs and other tools. Brahms software, represented by the ellipses, is running at the location of the RST, in the hab, on board the ERA, and on board one or more ATVs.

Brahms is a multiagent system, in which the fundamental processes are "agents" organized into "groups"; agents have "beliefs" and "activities" (see Clancey, in press for comparison to task modeling). Each Brahms system, running on these four (or more) platforms, is configured with "personal" agents, representing the interests and activities of corresponding people and systems. For example, RST Brahms includes the agents for the RST team; ERA Brahms includes an agent representing the ERA rover.

Each Brahms system in the MAA includes two other basic kinds of agents:

1. **Communication Agent:** People and objects in the real world interact with a Brahms agents through a communication agent (see Figure 1). The Dialog Agent is a type of communication agent allowing humans to have *conversations* with other mobile agents.
2. **Proxy Agents:** Represent agents and objects residing in other Brahms models; e.g., Hab Brahms includes an ERA proxy agent. Local Brahms agents interact with proxy agents in order to get or give information, as well as to request or promise actions. Proxy Agents allow for a seamless interaction between mobile agents even when a mobile agent is out of communication range to interact dynamically with the rest of the agents in the MAA.

The architecture is integrated through OAA messages, KAoS (Bradshaw et al. 1997), and CORBA:

- An agent in one Brahms model communicates with its proxy agent representation in another model through the KAoS middleware layer.
- Because continuous communication cannot be guaranteed, proxy agents represent the current state of other models, and may trigger recovery procedures if other agents are off line.
- Communication agents serve as interfaces between Brahms and external machines (e.g., a camera) using CORBA objects; as a special case, dialog agents interface between Brahms and the RIALIST speech system.
- Both communication agents and dialog agents are Brahms external agents (interfacing with Brahms through the Brahms API). For example, a Communication Agent is needed on the ATVBrahmsModel to interface with the MEX agent (for GPS tracking, communication equipment status, etc.).

The entire system shown in Figure 1 is first simulated in Brahms. In addition to the components shown in the ellipses, the people and external systems (e.g., ERA) are also simulated. This simulation therefore includes:

- Simulated People
- Simulated Systems (e.g., ATV)
- Software Agents (runtime)
  - Personal Agents of people (e.g., RST agents) and systems (e.g., ERA Agent, ATV Agent, Dialog Agent)
  - Communication agents
  - Proxy agents

Field Tests

The first systems integration field test was completed at NASA/Ames in April 2002. A "level 0" scenario integrated the RIALIST speech dialogue system and a distributed Brahms model running on the MEX ATV and a simulated ERA robot, using the MEX wireless network. The next field test was scheduled at JSC at the end of May, to include the ERA robotic control software running onboard the actual ERA robot.

The level 0 scenario demonstrated remote commanding of the ERA by the EVA astronaut, using human speech. The command to the ERA involves taking a still-picture of the astronaut ("ERA take a picture of me"). The speech dialogue system translates the human speech and communicates the derived command to the astronaut's personal Brahms agent. The astronaut's personal agent derives the correct robot command and communicates it to the ERA's personal agent running onboard the ERA (over the MEX wireless network), which in turn communicates with the ERA using a CORBA interface. The ERA executes the command and stores the image. The ERA's personal agent composes and stores the context of the picture in the form of a picture meta-data object and informs the EVA astronaut that the image has been taken, using the speech dialogue system.

Besides testing the RIALIST/Brahms/MEX interface, another objective of the field test was to apply a "simulation to implementation" methodology using a Brahms simulation model. This Use-Case method (Jacobson 1994) uses a Brahms simulation model as an agent- and object-based computational functional design of the overall software and hardware system. The runtime system replaces the simulated people and systems by their corresponding actual entities. This approach allows us to test a complete MAA system by simulating those elements that are not yet implemented or included in a field test (e.g. the robot).

Acknowledgments

Funding for this work is provided in part by the NASA/Ames Intelligent Systems Program, Human-Centered Computing area, managed by Mike Shafto.

References

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

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

Clancey, W. J. (in press). Simulating activities: Relating motives, deliberation, and attentive coordination. Cognitive Systems Research. Special issue on situated and embodied cognition.

Jacobson, I. 1994. Object-Oriented Software Engineering: A Use Case Driven Approach, Addison-Wesley Publishing Company, Reading, MA.