The Mobile Agents Integrated Field Test: Mars Desert Research Station April 2003

William J. Clancey[1], Maarten Sierhuis, Rick Alena, Sekou Crawford, John Dowding, Jeff Graham, Charis Kaskiris, Kim S. Tyree, and Ron van Hoof

NASA/Ames Research Center
Computational Science Division, MS 269-3
Moffett Field, California 943055
William.J.Clancey@nasa.gov

Abstract

The Mobile Agents model-based, distributed architecture, which integrates diverse components in a system for lunar and planetary surface operations, was extensively tested in a two-week field "technology retreat" at the Mars Society's Desert Research Station (MDRS) during April 2003. More than twenty scientists and engineers from three NASA centers and two universities refined and tested the system through a series of incremental scenarios. Agent software, implemented in runtime Brahms, processed GPS, health data, and voice commands—monitoring, controlling and logging science data throughout simulated EVAs with two geologists. Predefined EVA plans, modified on the fly by voice command, enabled the Mobile Agents system to provide navigation and timing advice. Communications were maintained over five wireless nodes distributed over hills and into canyons for 5 km; data, including photographs and status was transmitted automatically to the desktop at mission control in Houston. This paper describes the system configurations, communication protocols, scenarios, and test results.

Background

The Mobile Agents project anticipates exploration of Mars, in which a crew of six people are living in a habitat for many months. One long-term objective is to automate the role of CapCom in Apollo, in which a person on Earth (in Houston) monitored and managed the navigation, schedule, and data collection during lunar traverses (Clancey, in press b). Because of the communication time delay this function cannot be performed from Earth during Mars exploration, and other crew members will often be too busy with maintenance, scientific analysis, or reporting to attend to every second of a four to seven hour Extra-Vehicular Activity (EVA).

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

- Brahms Project Group (NASA-Ames: W.J. Clancey, Principal Investigator; M. Sierhuis, Project Manager; R. van Hoof, lead programmer; C. Kaskiris, modeler)
- RIALIST Voice Commanding Group (RIACS: John Dowding)
- MEX Vehicle & Wireless Communications Group (Ames: Rick Alena, John Ossenfort, Charles Lee)
- EVA Robotic Assistant Group (NASA-JSC: Jeff Graham, Kim S. Tyree (née Shillcutt), Rob Hirsh, Nathan Howard)
- Space Suit Biovest (Stanford: Sekou Crawford, in collaboration with Joseph Kosmo, JSC)
- NASA-Glenn Research Center/NREN satellite communications (Marc Seibert).

We have previously described how the Brahms simulation system (Clancey et al. 1998; Sierhuis 2001) has been adapted to provide both a tool for specifying multiagent systems and an implementation architecture for runtime agent interactions on mobile platforms (Clancey, et al., 2003). We have described how Brahms is used to model and control system interactions, and outlined two preliminary field tests at Johnson Space Center and Meteor Crater (September 2002). We presented a summary of advantages and limits of the Brahms architecture for multiagent applications. We have emphasized that building a practical system in a difficult terrain prioritizes issues of network robustness and diminishes, at least initially, theoretical questions about agent competitiveness and cooperation.

Mobile Agents Configuration

In an AI system, computational models make "intelligent" operation possible. The models in the Mobile Agent architecture include:

- Agents representing people in the simulation system (used for testing the design protocols)
- Models of devices (e.g., camera)
- Dynamic location model, including each agent and object (in terms of "areas" such as a habitat, and then specified by the Lat/Long coordination system)
- Network connectivity model, distributed in design of Comm and Proxy agents (which relate external devices and agents to a local platform)

[1] Also Institute for Human and Machine Cognition, UWF, Pensacola, FL.

- EVA Activity Plan: Sequence of activities specifying start and stop coordinate locations, a duration, and thresholds allowed.
- Language model: Word models and mapping of phrases to commands (with agent, location, object parameters)
- Command semantics, distributed in agent interactions, constituting a work flow for communicating, accessing, and storing data (e.g., photographs, coordinates, biosensors)
- Alarms, representing data thresholds (e.g., expected length of an activity within an EVA) and actions (e.g., where to communicate this information).
- ERA plans, locations, and control procedures (e.g., to take a photograph of a location)

In preparation for the April 2003 test, the project team developed three Brahms models to run respectively on laptops located on the EVA Robotic Assistant (ERA; Burridge & Graham 2001; Shillcutt et al. 2002), on an ATV, in spacesuit backpacks (for two astronauts), and in the Mars Desert Research Station (MDRS) habitat ("HabCom"). The HabCom model is entirely new, to monitor the EVA activity and biosensors.

Tests were performed at Ames in early March, first by wiring the laptops, and then with the wireless data network (MEX/KaoS) linking all components. The biosensors are wired to an iPaq PDA worn by the astronaut, which transmitted by bluetooth to a Minibook computer attached to the top of backpack. A GPS unit, camera, and headphone-microphone were all connected to the Minibook. The ERA Brahms controlled the ERA's camera through an API to the ERA's computer.

Changes were made after September 2002 (Meteor Crater) to handle key problems:

- **Agent architecture:** Cope with a brittle wireless network, with methods for handling lack of communication, as well as means for people to monitor agent status.
  - Used NetPhone to allow communications between support personnel at EVA site and HabCom
  - Implemented AgentMonitor display to allow HabCom to view entire state of every agent on the network (running on remote platforms)
  - Separated low-level sensor processing on iPaq from interpretation and data archiving on AstroBrahms
  - Implemented a rudimentary "forgetting" operation in Brahms, so modeler can deliberately delete beliefs that represent transient states (e.g., communications from other agents after they have been processed).
- **Hardware:** Sensors to indicate remaining power and provide warnings—short-term solution is running remote monitoring of Minibooks from laptop at ATV; bandwidth interference and microphone sensitivity—resolved by using two wired headsets for voice communications and recognition system; spacesuit dome greatly reduced noise; improved discipline for configuring connectors, by using dedicated kits; did not augment ERA capabilities.
- **Logistics:** Eliminated the pressurized spacesuit to focus on MAA infrastructure; in-situ tests were carefully staged over two weeks to meet objectives; MDRS provided a permanent field shelter for working in raining, cold, windy conditions (also sometimes dry and hot).

Comparison to recommendations in the Flairs 2003 report will show substantial progress was made. With the ability to test the system more thoroughly, yet more challenges were discovered.

The following is a summary of functionality implemented for April 2003 (* indicates new functions, mostly handled by the HabCom personal agent):

- **Location Tracking**
  - GPS mounted on *backpacks & ATVs
  - Flexible logging interval
  - *Sent to Remote Science Team (RST) at variable intervals
  - Naming from predetermined vocabulary
- **Science Data Logging**
  - Sample bags associated with locations
  - *Voice annotations associated with sample bags & locations
  - *Photographs logged by time & place
  - *Stored at hab & transmitted to RST
- ***Biosensor Logging**
  - Transmitted via iPaq to backpack
  - Logged at intervals & interpreted
- ***Activity Tracking**
  - Indicate start from predetermined list
  - Modeled by location & duration, sequence
  - Alerting for exceeding thresholds
- **ERA Commanding**
  - Move, follow, take a picture

For each of these functions, language was developed to support natural spoken phrasings in the dialog system. The essential expressions are:

start tracking my location every <N> seconds
start tracking my biosensors every <N> seconds
start <activity name> **activity**
call this location <location name>
where is [<location name> | Boudreaux]?
Boudreaux take a picture of [tracked astronaut | <location name>]
record/play {a} **voice note** {associated with <location name> | sample bag #}
create sample bag #
associate {this} [**voice note|sample bag** {#}] **with** <location name>
how much time is left {until next activity}?
upload [all | one] image{s}

PEOPLE	Microphone & Headphone	PERSONAL AGENTS	Comms Agents	Ext. Systems
Astro1	←voice→	**$Agent_{Astro1}$**	←API→	GPS, ERA Biosensors Camera
| radio |		| wireless network |		
HabCom	←voice→	**$Agent_{HabCom}$**	←API→	Email (to RST)
| radio |		| wireless network |		
Astro2	←voice→	**$Agent_{Astro2}$**	←API→	GPS, ERA Biosensors Camera

Figure 1. Simplified schematic of MA configuration April 2003: Astronauts and HabCom communicate with via radio (not all links are shown); people speak to their Personal Agents (Pas) on local computers using a microphone and receive feedback on headphone. PAs are communicating locally with external systems via "communications agents," providing an Application Programming Interface (API) to read data and control devices (e.g., camera). PAs (all implemented in Brahms on different computers) communicate via a wireless network (with repeaters) using KAoS "agent registration" system.

Table 1: Field test work day on which milestone was accomplished with problems discovered

Field Test Milestone	Day	New Problems Discovered
Wired test inside MDRS	3	Heat burned power regulator; voice commands misinterpreted
Wireless test inside	4	Alarms & email bogging down system
Test porch and walk inside	5	Many command processing bugs/incomplete code
Front Porch with GPS & Walk around MDRS	6	Initialization logic: No GPS inside, can't start biosensors until headset/helmet on
Repeat MDRS Walk with ERA	7	Basic command workflow still being tested first time
Pedestrian EVA	8	Multipaths from hills, lose network
Pedestrian w/ERA	9	Unreliable ERA navigation
Lith Canyon EVA	11	System tolerant to node reboot & network dropout, but topography-specific limits

Six scenarios were designed, ranging from a simple walk around the hab to a full-day drive onto a plateau. Actual testing involved incremental, pre-planned stages:

1) Wired test in the lower deck of MDRS to confirm communications protocols and peripheral connections
2) Wireless test inside the hab (without full suits, emphasizing communications and biosensors)
3) Test standing on the front porch of MDRS (allows use of GPS for first time)
4) Full walk (over one hour) around MDRS, following a script to test basic functionality; all systems running except ERA.
5) "Pedestrian EVA" with ERA, walking from porch to a dry wash about 100 meters south, gathering samples, taking photos, commanding ERA to take a photo, and return
6) "Lith Canyon EVA" involve two repeaters, ATV providing gateway to a LAN in the canyon, and a hour or more of scripted sampling and photography.

Table 2*: Lith Canyon Scenario script (first half) provided to Astronauts.* Activity plan key: (Duration, Duration threshold in minutes, Distance threshold in meters)

ACTIVITY PLAN	GEOLOGISTS' SCRIPT
<EVA PREP> (-,-,-)	1. Drive in EVEREST with backpacks, helmets, suits, all equipment 2. Start Minibooks & GPS 3. Don suit with boots, gaiters, radios & headsets 4. Put on backpack & helmet & connect cables
Checking equipment (20, 5, 20)	1. "Start CHECKING EQUIPMENT activity" 2. "Start tracking my location every 60 seconds" 3. "Start tracking my biosensors every 5 minutes" 4. "Start WALKING TO TOP OF CANYON activity"
Walking to top of canyon (10, 0, 10)	{Astronaut 2 improvised a voice note during the walk}
Sample fossils (10, 5, 0)	1. "Start SAMPLE FOSSILS activity" 2. Sample bag, voice annotation, association, photo 3. "Start WALK TO HEAD OF CANYON activity"
Walk to head of canyon (10, 0, 10)	< Walk carefully down the hill and proceed to the head of the canyon to the south (your left) >

We arrived Saturday, March 30, and completed setup of all equipment in and around MDRS on Sunday. First tests began Monday morning. The scenarios (already pruned to eliminate three EVAs to remote sites), were accomplished very gradually (Table 1). The day count subtracts one for Sunday April 6, a rest day. Nearly 5 of 11 work days were devoted to model modifications and testing. Functionality

errors stem from incomplete end-to-end simulation (caused by inadequate resources and planning).

The astronaut-geologists were provided with scripts to indicate the sequence of activities, including locations, and requested or optional commands to test. The astronauts could skip or repeat activities by indicating what they were doing (subject to predefined location and timing constraints). This was most useful for the more realistic exploration at Lith Canyon, where Astro2 improvised a voice annotation (Table 2).

Figure 2 shows the topography configuration for Lith Canyon; it required nearly two days to deploy communication equipment in preparation for this EVA.

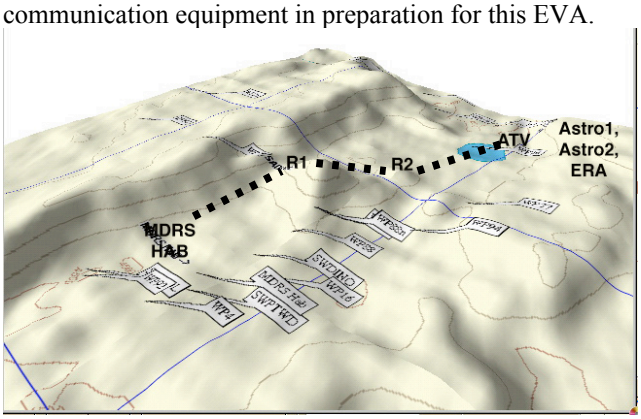

Figure 2. Topographic layout of MDRS (hab), Cisco repeaters, ATV LAN gateway, and astronauts with ERA in Lith Canyon (about 5 km from MDRS).

Table 3: Lith Canyon Communications Processed by Personal Agents

Communication	Astro1	Astro2	HabCom
broadcast_alert	2	0	4
create_location	1	1	0
download_images	3	1	0
GPS	3	2	0
GPS_start_logging	1	2	0
initialization	2	4	4
location_with _samplebag	1	1	0
newData	0	0	9
new_activity_started	2	1	7
SampleBag	3	1	0
start_specified_activity	2	0	2
storeData	27	13	0
TakePicture	0	1	0
voice_annotation_with _location	1	1	0
voice_annotation_with _samplebag	1	1	0
VoiceAnnotation	7	1	0

Lith Canyon Relay Test

The Lith Canyon site involves broken ledges and steep cliffs 5 km from MDRS. The topography caused several serious problems:

- A wireless "shadow" occurred at the head of the canyon (as expected), causing the computerized backpacks to drop out of the network linking the astronauts back to MDRS. (Communications were properly queued and handled when possible.)
- The astronauts were unwilling to pass over a meter-high drop off in the canyon, requiring them to change the plan and walk around.
- The ERA was unable to follow the astronauts into the canyon because of the terrain, and even along the ledge had to be directly teleoperated with a person standing nearby.

The Lith Canyon field test was a major accomplishment for the Mobile Agents project. The geologists' backpack computers running Brahms were wirelessly networked to another computer on an ATV 75 m away on a ledge across the canyon, and from there to a laptop running in MDRS more than 5 km away (Figure 2). The EVA lasted about 1.5 hours, as batteries allowed.

Table 3 shows communications that were generated by personal agents during the canyon test. In addition, one command was given to the ERA PA by the Astro2 PA (Astro1 asked the ERA to take a picture; however it was not executed.)

An *explicit confirmation* mode was added to the dialog system, requiring each command to be confirmed by the astronaut prior to execution (e.g., "Do you want to start walk to head of canyon activity?"). Most commands are forwarded to the MDRS computer (HabCom) and then responses returned to the astronaut agents, causing the dialog system to generate a verbal feedback (e.g., "Activity walk to head of canyon activity started"). This confirmation and feedback protocol was implemented through trial and error during the first week of the field test. It ensures that the correct command is being processed and that it has been executed. Although seeming obvious now, use of the system in context was required to discover what kind of feedback was required.

Analysis and Conclusions

Field test results can be summarized according to lessons learned about the hardware, the agent architecture, and logistics of setting up the system and carrying out the scenarios:

- **Hardware**
  - Technology required for field science is strongly topography driven
  - A robot should be capable of working in terrain that geologists explore (e.g., tractors/spider legs)
  - Need automated antenna and video tracking
  - Need faster computers all around
  - Adapt ERA's differential GPS for astronauts

- **Agent Architecture**
  - Copes well with loss of signal; but Astro PAs must take over some HabCom monitoring functions
  - Need assertions to verify end-to-end functionality (e.g., has photo arrived at RST?)
  - Must integrate speech output, astronaut voice, and HabCom for recording
  - Astronauts work in parallel, lack voice loop; they must coordinate to avoid work redundancy
- **Logistics**
  - Need formal data network specification (GPS, computer, radio, biosensors)
  - Need written specs/deliverables (not just design documents)
  - Need field backups for all computers

Ideally, mobile agents should run for the duration of an entire mission—perhaps several years. However, agents interactions with their environment result in an ever expanding belief set (memory set). Not managing memory results in serious performance degradation of the agents' reasoning state network. An interim solution, implemented during the field test, is to retract transitory communication beliefs (i.e., that a request has been received). A more systematic, built-in model of forgetting is required.

In this early stage of development, the HabCom person was responsible for monitoring the various Brahms systems. He listened and responded to: the voice loop, information spoken by his PA (such as alerts), and field radios (or IP phones) to verify system responses. Because all three require a different focus, he frequently missed problems and requests. This discovery exemplifies our approach of "empirical requirements analysis." One may design clever agent algorithms and architectures, but in practice one will find that simple services are needed that were never considered back in the home lab. In particular, we are frequently discovering new tools required to monitor and verify the system's operation during this developmental process. This phase can be expected to last many years, and involves viewing the agents as assisting in the research and ongoing redesign of the system.

Brahms was not designed from the ground up to be a distributed, highly available mission support system. Persistence and failure recovery are some aspects that require near-term redesign. Additional requirements may need to be imposed on the Brahms language. For example, better methods are required to allow agents to recover from system failures, multi-task, inspect their own state and adjust themselves in different contexts. Methods must be improved for handling large volumes of data that may need to be stored, but only sampled and interpreted periodically.

We conclude that the following aspects of the MA design and field test worked especially well and were crucial to our progress:

- **Human-Centered Design (Technology Pull)**
  - Authentic work scenarios (real geologists doing real work)
  - Analog simulations during MDRS5 (Clancey 2002a) plus historical analysis of Apollo CapCom (Clancey in press b)
  - On-site requirements analysis (MDRS5) -> Voice command design -> Simulation in Brahms -> Distributed Implementation
- **Technology Retreat Facility**
  - Attractive, isolated, evocative setting
  - Utilities augmented: Broadband ISP, LAN, toilets
  - Nearby inexpensive motels & restaurants
  - Resident handyman
- **Management Structure**
  - Commander plus subsystem point-of-contacts
  - Realistic 9 am–5 pm schedule
  - Required 9 am briefing; replanning at end of day
  - Arrive Friday, start Monday am, Sunday off, cleanup second Saturday (11 work days)

As suggested by the previous tests, we conclude that a multiagent simulation with scenario-based formal specification *accelerates cross-institution collaboration* for integrating sensors, automation, procedures, and communications.

We plan to return to MDRS in April 2004 with these objectives:

- Complete the Pedestrian and Lith Canyon scenarios
- Extend navigation & schedule monitoring
- Develop a medical agent to interpret biosigns
- Develop a mission console for HabCom to log alerts
- Provide HabCom with a map of locations for all people and agents, including their activities
- Add science database, with RST access for shared annotation
- Plan and track multi-day EVAs with the RST

Discussion of Related Work

Here we compare the MAA with some other multiagent frameworks that focus on team coordination.

The Teamcore agent framework (Pynadath and Tambe 2003) is similar to the MAA in the sense that it allows new or existing agent programs that have no pre-existing coordination capability to work together. While in the MAA the PAs provide mobile agents with the capabilities to work together performing team tasks, in Teamcore this function is provided by "proxies." In the MAA *proxy* refers to agents that represent PAs in agent environments running on different machines. Proxy agents in MAA thus represent the PAs in other agent systems, whereas Teamcore proxy agents represent agents that can be written in different languages with a Teamcore wrapper to allow these agents to communicate using the KQML agent communication language (Finin et al. 1997).

Another similarity between Teamcore and MAA is the need to facilitate human collaboration. Our objective is to allow an EVA crew to work together (including the crew in the habitat and remote science teams on Earth, as well as with robots and other science tools and devices). The Teamcore infrastructure has been applied to *a computer*

simulation of the evacuation of civilians from a threatened location and to assist the Teamcore's research team in their routine coordination (e.g., scheduling team meetings). In this second application, Teamcore is also addressing the mobility issue of people by integrating GPS devices and PDAs. Differences between the MAA and Teamcore include: 1) MAA allows the human user to dynamically create new locations; 2) agents in the MAA are mobile, therefore the architecture and agent design have to deal with the fragility of a wireless communication network; and 3) MAA provides assistance to people coordinating with robots, external devices, and medical monitoring.

RETSINA is a multiagent system (MAS) infrastructure that allows developers to implement large distributed societies of software agents (Sycara et al. 2003). RETSINA is an agent infrastructure, not an agent language. Research with RETSINA focuses on understanding what is needed to provide a "domain independent and reusable substratum on which MAS systems, services, components, live, communicate, interact and interoperate, while the single agent infrastructure is the generic part of an agent that enable it to be part of a multiagent society" (Sycara et al. 2003). We also intend for the MAA to be a MAS infrastructure that allows independent agents (people, software agents, robots and devices) to be part of a large multiagent society, or even multiple societies–including EVA and habitat agents, and RST societies (science, medical, engineering, mission control teams).

One key difference between RETSINA and Mobile Agents is the domain: The vision of the RETSINA team is the internet's computational world, populated with agent societies. In the current MAA, agents are not directly connected to the internet, but run on a dedicated wide-area wireless communication network. Connection to the internet is currently provided by a single agent that can e-mail to members of an RST. Sycara, et al. (2003) provide a definition of the MAS infrastructure service layers. We aim to provide many of these services in the MAA as well. To this extent we are hoping to use existing systems, and perhaps in the future we will use some of RETSINA's capabilities within the MAA. Currently, some of the MAA infrastructure is provided by KAoS (Bradshaw et al. 1997), such as the communication infrastructure, name to location mapping, and some of the multiagent management services. We are also currently integrating Brahms with the policy capability of KAoS (Bradshaw et al. 2003) to handle agent interactions when communication breakdowns require local agents to assume responsibilities, and then relinquish them when the network is re-established.

Acknowledgments

Funding is provided in part by the NASA/Ames Intelligent Systems Program, Human-Centered Computing area, managed by Mike Shafto. Our team includes a dozen other Brahms, ERA, and communications specialists. For related information (including daily field reports from MDRS16), see http://bill.clancey.name, http://www.marssociety.org, and http://www.agentisolutions.com.

References

Alena, R., Ossenfort, J., Lee, C., Walker, E., Stone, T., in press. Design of Hybrid Mobile Communication Networks for Planetary Exploration. *IEEE Aerospace Conference 2004*.

Bradshaw, J. M., Dutfield, S., Benoit, P., and Woolley, J. D. 1997. KAoS: Toward an industrial-strength generic agent architecture. In J. M. Bradshaw (ed.), *Software Agents*, 375-418. Menlo Park, CA: AAAI Press.

Bradshaw, J. M. et al. 2003. Representation and Reasoning for DAML-Based Policy and Domain Services in KAoS and Nomads. *AAMAS '03*, Melbourne, Australia.

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

Clancey, W. J. 2002a. A Closed Mars Analog Simulation: the Approach of Crew 5 at the Mars Desert Research Station. *Mars Society Annual Conference*, Denver.

Clancey, W. J. 2002b. Simulating activities: Relating motives, deliberation, and attentive coordination. *Cognitive Systems Research*, **3**(3) 471-499.

Clancey, W. J., Sierhuis, M., van Hoof, R., and Kaskiris, C. 2003. Advantages of Brahms for Specifying and Implementing a Multiagent Human-Robotic Exploration System. *Proc. FLAIRS-2003*, 7-11.

Clancey, W. J. in press a. Roles for Agent Assistants in Field Science: Understanding Personal Projects and Collaboration. *IEEE Transactions on Systems, Man and Cybernetics*. Issue on Human-Robot Interaction.

Clancey, W. J. in press b. Automating CapCom: Pragmatic Operations and Technology Research for Human Exploration of Mars. In C. Cockell (ed.), *Martian Expedition Planning.* AAS Publishers.

EVA Robotic Assistant. URL http://vesuvius.jsc.nasa.gov/er_er/html/era/era.html.

Finin, T., Y. Labrou, and Mayfield, J. 1997. KQML as an agent communication language. In J. M. Bradshaw (ed.), *Software Agents*, 291-316. Menlo Park, CA: AAAI Press.

Pynadath, D. N., and Tambe, M. 2003. An Automated Teamwork Infrastructure for Heterogeneous Software Agents and Humans. *Autonomous Agents and Multi-Agent Systems*, 7, 71-100.

Shillcutt, K., Burridge, R., Graham, J. 2002. Boudreaux the Robot (a.k.a. EVA Robotic Assistant). *AAAI Fall Symp. on Human-Robot Interaction*, Tech Rpt FS-02-03, 92-96. Menlo Park, CA: AAAI Press.

Sierhuis, M. 2001. *Modeling and simulating work practice*. Ph.D. thesis, Social Science and Informatics (SWI), University of Amsterdam, The Netherlands, ISBN 90-6464-849-2.

Sycara, K., Paolucci, M., Velsen, M. v., and Giampapa, J. 2003. The RETSINA MAS Infrastructure. *Autonomous Agents and Multi-Agent Systems*, 7, 29-48.