PerAdaMagazine
towards pervasive adaptation
10.2417/2200910.1823

NASA deploys intelligent software agent technology

Maarten Sierhuis, William J. Clancey, and Timothy Hall

A new automated flight-control system for the International Space Station reduces workload by 90% by employing ethnographic observation and agent-based modelling and simulation.

We have used work-practice analysis and design, combined with a 'from simulation to implementation' agent-based software-engineering approach (see Figure 1), to improve International Space Station (ISS) flight operations in NASA's Mission Control Center (MCC).[1] Our approach is based on the Brahms agent-oriented programming language and agent-based modelling and simulation environment,[2–4] developed over many years in the Intelligent Systems Division at NASA Ames Research Center (ARC).[5]

The ISS astronaut crew is supported around the clock by a variety of specialists, including flight controllers, physicians, planners and engineers (see Figure 2). In addition to verbal communication and direct control of ISS systems, the most important means of interaction between crew and ground team is through exchange of more than 30 types of files, including crew-timeline updates, crew email, photographs and exercise logs. A dedicated flight-control position in the Johnson Space Center (JSC) MCC serves as the 'switchboard operator' for file transfer between the astronaut crew's network of laptops and ground-support teams.

During the more than 10 years that the ISS has been assembled, the MCC—not unlike many corporate information-technology environments—has seen introduction of new networks, as well as the evolution of legacy systems and changes to network-security policies. This dynamic environment has led to development of many different methods to transfer data, receive information and distribute files. Unfortunately, this has resulted in the need for a dedicated 'switchboard operator'—an orbital communication adapter (OCA) officer—where a skilled engineer is assigned to the mundane task of file distribution and management. The operator uses a combination of manual and electronic transfer methods (such as folder drop boxes, email and electronic flight notes) to receive and forward files to the astronaut crew. Now, as the ISS assembly nears completion, much of this work has become routine and can be automated using intelligent software agents that integrate a variety of applications on different networks.

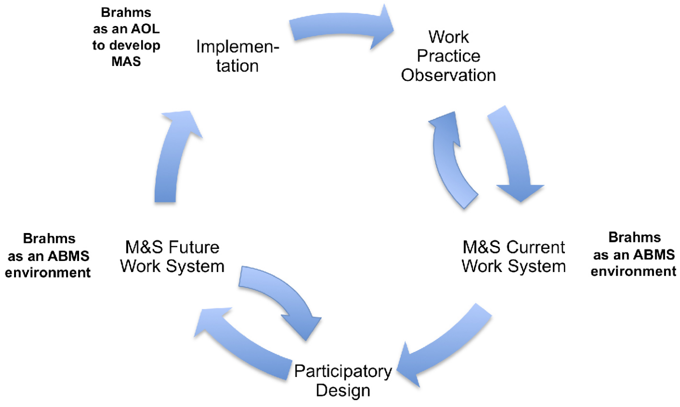
Brahms as an AOL to develop MAS
Implemen-tation
Work Practice Observation
Brahms as an ABMS environment
M&S Future Work System
M&S Current Work System
Brahms as an ABMS environment
Participatory Design

Figure 1. *From simulation to implementation. AOL: Agent-oriented (programming) language. ABMS: Agent-based modelling and simulation. MAS: Multi-agent system. M&S: Modelling and simulation.*

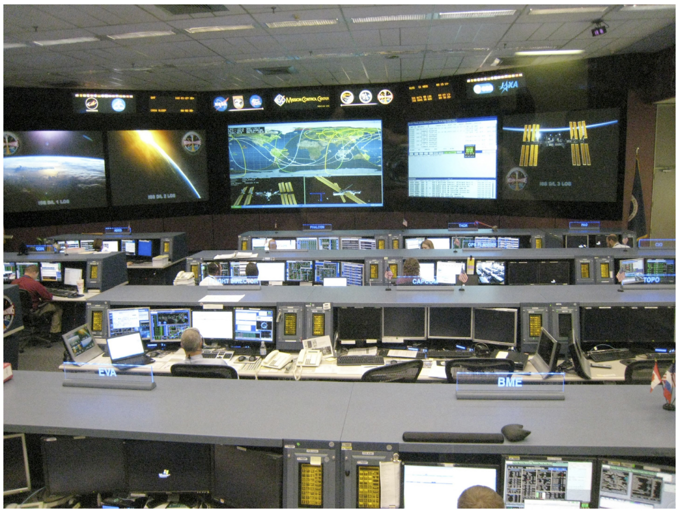

Figure 2. *The International Space Station's Mission Control Center.*

Figure 3 shows part of the simulated OCA officer's activities timeline for orbit 3 (the third shift). At top right (1), the

Continued on next page

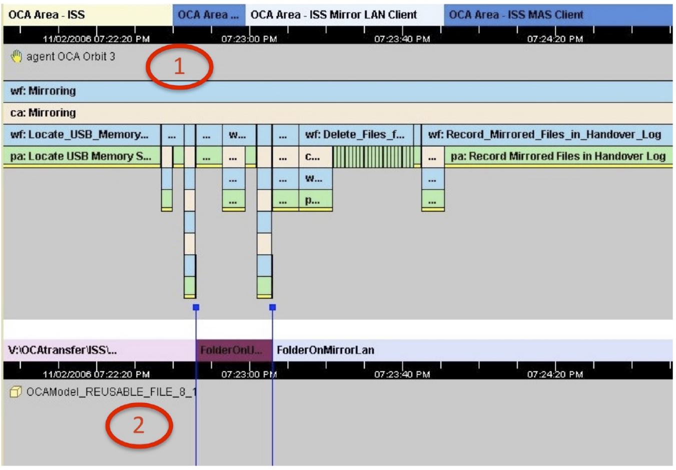

Figure 3. Simulation of orbital communication adapter (OCA) officer's manual mirroring activity—using the universal serial bus 'sneakernet' method—during orbit 3. wf: Workframe. ca: Communication activity. pa: Primitive activity. The encircled numbers refer to the order in which the panels are discussed in the text.

mirroring activity and subactivities are executed. The timeline also shows the actual file object being moved (2) from folder `V:` to a folder on the mirror local-area network using a USB 'stick.' The coloured bars above the black timeline show location movements of the OCA officer (1) and file object (2).

The simulation-to-implementation methodology for developing the OCAMS (orbital communication adaptor mirroring system: see Figure 4) software uses simulation to design the system and quantitatively evaluate and test its performance on actual data prior to implementation (see Figure 1). This method began with a partnership between the OCA officers and artificial-intelligence experts. ARC researchers observed OCA operations in the MCC backroom and then collaboratively simulated the manual mirroring operations in Brahms.[6] Brahms agent models describe how people do their work, chronologically, including a model of the office environment. They also specify the workflow tools they use and how people communicate (e.g., by telephone, voice loop or email).[7]

Using Brahms' activity-based modelling, the joint JSC-ARC team quantified the cost or effort involved in manual mirroring.[8] Subsequently, this operations simulation was directly edited to create a 'future-operations' simulation, in which the model of what the OCA officer did manually was converted into automated software agents. We performed a validation of the future-operations simulation against historical logs of the OCA officer's file-handling actions and produced metrics on the cost or effort savings that could be expected. We then extracted the agents from the simulation and converted them into the OCAMS standalone software package.

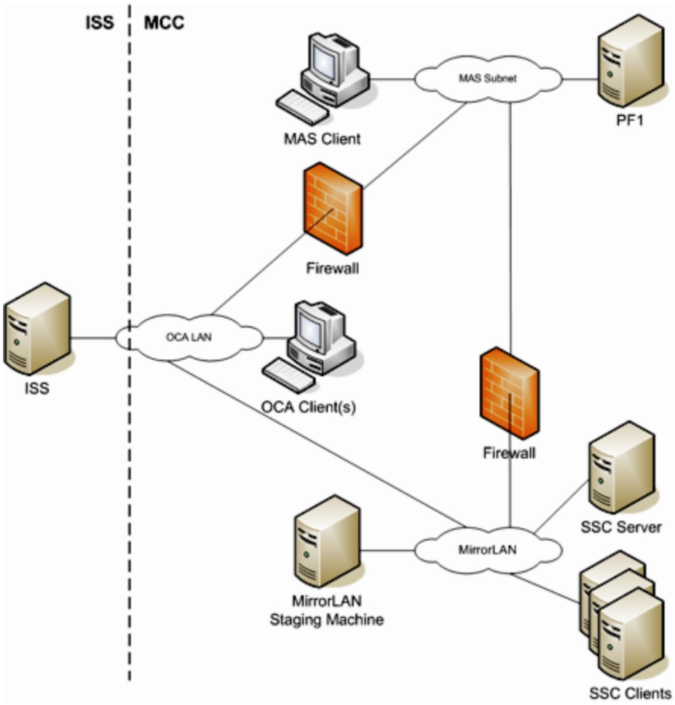

Figure 4. OCAMS initial implementation runs on the mirror local-area-network (LAN) staging machine, transferring files between the OCA client computers and the space-station computer (SSC) server that replicates a subset of the ISS file system. PF1: File server in the MCC.

By September 2008, all OCA officers used the initial release of the OCAMS software. It was estimated to save about 5–10% of each total shift time. This was the first step towards allowing the backroom flight controllers to focus on more complex priority tasks and less on simple file and data management. The automated logging feature alone was especially welcome for eliminating the tedious process of recording the mirroring (to maintain a mirror of the ISS system on the ground), which itself already duplicated the work they had performed on the ISS computers. By early 2009, the OCAMS package included an editor for modifying file-handling rules, enabling OCA officers to change OCAMS operations without engaging developers to reprogram the software agents (e.g., to change naming conventions or where files are stored). During 2009, archiving and delivering files (e.g., providing exercise logs to the biomedical

Continued on next page

engineers) was also automated, further reducing the OCA officers' workload by approximately 25–30%. This enabled the OCA management to combine the position with another backroom flight-control position, literally reducing the labour required to support ISS operations. Following an OCA equipment upgrade scheduled for late 2009, OCAMS Revision 3 agents will be distributed, allowing the file-handling process to be monitored and controlled when necessary from remote-console locations, like the OpsPlan flight-controller position in the main MCC flight-control room (or 'front room').

The final stage of automation planned for 2010 will automate most up- and downlink operations to ISS, reducing the OCA officers' workload by 90%, essentially eliminating the need for a dedicated OCA officer position in the MCC and freeing up the time of skilled engineers to focus on the more abstract tasks involved with planning ISS operations. The remaining 10% of OCA work will focus on special file-transfer requests that require supervisory approval or manual overrides that will be managed with available bandwidth from other positions.

In summary, OCAMS' simulation-to-implementation multi-agent-system development methodology combines ethnography, participatory design,[9] multi-agent simulation and agent-based systems integration to accurately model, simulate and then automate tasks. The methodology yields a hybrid simulation that combines actual and simulated systems (such as email) and objects (such as files) with simulated people. We used actual flight-operations data to validate the simulations and the prototype workflow tool.

More generally, the project demonstrates a systems-integration architecture in which software agents are used to link arbitrary hardware and software systems to distributed teams of people on earth and in space, a solution that will be needed for future human space flight. OCAMS may be the first step in developing an interplanetary multi-agent system!

We thank all OCA officers in Operations Division 4 at NASA JSC for their continued support in the use and development of OCAMS. We also thank our current funders at NASA JSC, and all past funders of the Brahms project.

Author Information

Maarten Sierhuis
Code TI, Collaborative Assistant Systems
NASA Ames Research Center
Moffett Field, CA
and Carnegie Mellon University Silicon Valley
Moffett Field, CA
http://homepage.mac.com/msierhuis

Maarten Sierhuis is a senior scientist at NASA ARC, adjunct professor at Carnegie Mellon's Silicon Valley campus and visiting professor in the Man-Machine Interaction group at Delft University of Technology (Netherlands). He has over 16 years of experience in agent-based simulations and multi-agent systems for mission operations and holds a PhD in social-science informatics from the University of Amsterdam (Netherlands).

William J. Clancey
Institute for Human and Machine Cognition (IHMC)
Pensacola, FL
and
NASA Ames Research Center
Moffett Field, CA
http://bill.clancey.name/

Bill Clancey is on an Intergovernmental Personnel Act assignment from the IHMC to NASA ARC, where he has been leading the effort in human-centred computing. He received a PhD in computer science from Stanford University in 1979 and has been involved in expert systems research in Stanford's Knowledge Systems Laboratory from the early days of the Mycin project in 1975. Clancey's work on 'conceptual coordination' provides a theoretical foundation for relating human cognitive and social processes to the mechanisms used in artificial intelligence and robotic systems.

Timothy Hall
Mission Planning and Software Integration Team, DO4
NASA Ames Research Center
Moffett Field, CA

Tim Hall is the group leader of the Mission Planning and Software Integration Team. His team develops the integrated mission plans for both the Space Shuttle and the ISS, as well as the

Continued on next page

software applications that the ISS crew use to execute the mission plans in orbit. He is part of a Mission Operations Directorate intercentre initiative to work with other NASA centres to leverage agency experience and expertise for future ISS, Constellation and Exploration programmes.

References

1. M. Sierhuis, W. J. Clancey, *et al.*, *Organization and work system design and engineering*, in L. Yilmaz and T. Oren (eds.), **Agent Directed Simulation**, Wiley. In Press.
2. http://www.agentisolutions.com Agent iSolutions. Accessed 24 September 2009.
3. M. Sierhuis, W. J. Clancey, *et al.*, *Brahms: a multiagent modeling environment for simulating work processes and practices*, **Int'l J. Simul. Process Modell. 3** (3), pp. 134–152, 2007.
4. M. Sierhuis, W. J. Clancey, *et al.*, *NASA's OCA mirroring system: an application of multiagent systems in mission control*, **Proc. Auton. Agents Multi-Agent Syst. (AAMAS)**, 2009.
5. http://ti.arc.nasa.gov/ Intelligent Systems Division at NASA Ames Research Center. Accessed 24 September 2009.
6. W. J. Clancey, P. Sachs, *et al.*, *Multi-agent simulation to implementation: a practical engineering methodology for designing space flight operations*, **Lect. Notes Artif. Intell. 4995**, pp. 108–123, 2008. 8th Annu. Int'l Worksh. Eng. Soc. Agents World (ESAW 07)
7. W. J. Clancey, P. Sachs, *et al.*, *Brahms: simulating practice for work systems design*, **Int'l J. Hum.-Comput. Stud. 49**, pp. 831–865, 1998.
8. M. Sierhuis, W. J. Clancey, *et al.*, *Brahms: an agent-oriented language for work practice simulation and multi-agent systems development in multi-agent programming*, M. D. Rafael, H. Bordini, J. Dix, and A. El Fallah-Seghrouchni (eds.), 2nd ed., **Multi-Agent Programming**, Springer, 2009.
9. J. Greenbaum and M. Kyng (eds.), **Design at Work: Cooperative Design of Computer Systems**, Lawrence Erlbaum, Hillsdale, NJ, 1991.

© 2009 PerAda