Modeling and Simulating Work Practice: A Method for Work Systems Design

Maarten Sierhuis, *Research Institute for Advanced Computer Science/Universities Space Research Association*

William J. Clancey, *University of West Florida and NASA Ames Research Center*

Modeling actual human activity requires understanding the effects of communication, collaboration, teamwork, tool and workspace usage, and problem solving and learning behavior. Using a work system design method that contains these variables can produce powerful insights into complex system interactions.

Work systems involve people engaging in activities over time—not just with each other, but also with machines, tools, documents, and other artifacts.[1] These activities often produce goods, services, or—as is the case in the work system described in this article—scientific data. Work systems and work practice evolve slowly over time. The integration and use of technology, the distribution and collocation of people, organizational roles and procedures, and the facilities where the work occurs largely determine this evolution.

Improving or redesigning systems is sometimes accomplished through a *business process reengineering* approach.[2] Business process reengineering is usually based on business process flow analysis, typically performed by business consultants who focus on work products. The result is usually an improvement involving technology, such as a workflow tool. We call this a *machine-centered* approach for work systems design because functional transformations are the focus of attention. However, focusing on the flow of products and data through a work system often ignores the way the people in the organization actually prefer to work.[3,4] As a result, engineers often receive requirements for new technology without a thorough understanding of how the newly designed system might affect human communication, collaboration, and workspaces, as well as problem solving and learning.

In this article, we present a *human-centered* work system design method based on modeling and simulating work practice—that is, what people actually do. Rather than abstracting human behavior as work processes or tasks—functional idealizations of the work to be accomplished—we model people's activities comprehensively and chronologically throughout the day.[5] We emphasize that an analysis of how work gets done must be open to understanding the effects of behaviors in different places and times, details often omitted in a product-oriented task analysis. For example, someone might not schedule meetings at the office before 10:30 a.m. because of a babysitter's schedule, or he or she might use scheduling software on a computer at home to reserve meeting rooms for later that day. Such practices are relevant to the design of workplace facilities and scheduling. We call our method human-centered because we focus on how people organize their work life and the details of their practices. We believe this best suggests work system transformations, including any different tools and processes that might eventually be required.[3] For a look at other relevant research, see the sidebar "Related Work."

The Brahms language: A model-based view of work practice

Most engineering disciplines have methods and tools to help in understanding complex system interactions; often such tools help engineers develop system prototypes. We have developed a tool called Brahms that supports our method for modeling and simulating work practice.[6,7] Brahms is a multiagent modeling and simulation environment for dealing with the complex human–machine system interactions.

In software engineering, *model-based* refers to a system design approach in which the system is described in terms of the structure and behavior of

Related Work

Several researchers have developed different work design approaches through qualitative work in interdisciplinary academic fields that combine social science with a system analysis perspective.[1,2] A wave of Scandinavian participatory-design projects conducted in the late 1980s partially stimulated this rush.[3] We have adopted two principles from the Scandinavian participatory-design approach: redesigning a work system requires understanding how the work is actually performed, and *participatory* means that the workers must participate as designers of their system.

This Scandinavian approach is essentially human-centered but does not generally involve formal modeling of human–system interactions, as in Brahms. For example, contextual design[4] is based on the development of mostly paper-based models of a work system.

The lack of a theory of work practice has hampered the development of a generic work system engineering approach because it makes work system design an art. What is needed is a theory and an associated method for developing formal models of work practice. This will bring a human-centered work system design approach into mainstream methods for technology development, especially software engineering. Such an approach facilitates design conversations, creating a desperately needed bridge between scientists, workers, and technology engineers.

We can use a formal model of people's work practice as a design model for how technology impacts the total system—a holistic system design approach. Developing formal models of work practice means we need to model people's behavior at the activity level. Relevant work in modeling and simulating human behavior comes from different scientific disciplines:

- The business process modeling and simulation community creates formal specifications of an organization's business processes.[5]
- Cognitive-modeling tools, such as Soar and ACT-R, incorporate a cognitive theory for predicting human mental processes in controlled laboratory experiments.[6,7]
- The field of distributed artificial intelligence developed the notion of using multiple agents in complex problem-solving tasks,[8] which is essential for modeling teams of people collaborating in organizations.
- Computational organization modeling tests theories of communication and optimal decision making in human organizations.[9]
- The computational-economics community uses an agent-based simulation environment (for example, Swarm) as a tool for studying economic theories in complex systems.[10]

Reference

1. J.M. Corbett, L.B. Rasmussen, and F. Rauner, *Crossing the Border: The Social & Engineering Design in Computer Integrated Manufacturing Systems*, K.S. Gill, ed., Springer-Verlag, London, 1991.
2. K.J. Vicente, *Cognitive Work Analysis: Towards Safe, Productive, and Healthy Computer-Based Work*, Lawrence Erlbaum Associates, Mahwah, N.J., 1999.
3. P. Ehn, *Work-Oriented Design of Computer Artifacts*, Arbetslivcentrum, Stockholm, 1988.
4. K Holtzblatt and H. Beyer, *Contextual Design: A Customer-Centered Approach to Systems Design*, Morgan Kaufmann, San Francisco, 1998.
5. R.J. Mayer et al., "Framework and a Suite of Methods for Business Process Reengineering," *Business Process Change: Reengineering Concepts, Methods and Technologies*, V. Grover and W.J. Kettinger, eds., Idea Group, Hershey, Pa., 1998.
6. A. Newell, *Unified Theories of Cognition*, Harvard Univ. Press, Cambridge, Mass., 1990.
7. J.R. Anderson and C.J. Lebiere, *The Atomic Components of Thought*, Lawrence Erlbaum Associates, Mahwah, N.J., 1998.
8. P.E. Agre, "Computational Research on Interaction and Agency," *Artificial Intelligence*, vol. 72, nos. 1–2, 1995, pp. 1–52.
9. K.M. Carley and M.J. Prietula, eds., *Computational Organization Theory*, Lawrence Erlbaum Associates, Mahwah, N.J., 1994.
10. F. Luna and A. Perrone, "Agent-Based Methods in Economics and Finance: Simulations in Swarm," *Advances in Computational Economics*, A.N. Hans Ammam, ed., Kluwer Academic, Norwell, Mass., 2002.

defined components.[8] In particular, to model the work practice of a human activity system, we must create a dynamic model that shows how the system changes over time. Figure 1 describes an operational method for developing a formal computational model and a simulation of a work practice for a human activity system. It also shows the relation between four methods for using Brahms in a modeling effort.

Method M1: Work practice analysis

The purpose of method M1 is to observe and analyze a human activity system.[9,10] The goal of the analysis is to gather useful data that informally describes work practice and then to create (with the Brahms language) a formal model of work practice in M2. M1 modelers should be workers from the work system, work system designers, and anthropologists.

Method M2: Formal model of the work practice

The purpose of method M2 is to formalize the informal data gathered during M1. In Brahms terms, this is where we develop the Brahms model. The formal-system modelers translate the informal models developed in M1. The formal modelers and the informal modelers do not necessarily have to be the same; in fact, the skill sets for these two types of modelers differ significantly. Formal Brahms modelers are usually people who understand the concept of agent-based modeling and simulation and often have experience in developing rule-based systems.

Method M3: Simulation

In Method M3, formal modelers run the Brahms simulator with the model as input and the work practice simulation as output. The M3 method is the Brahms compile-simulate-debug cycle, because the modeler will compile, simulate, and fix the errors in the formal model until the desired agent behavior is simulated.

Method M4: Observing the simulation

The purpose of method M4 is to observe and investigate the work practice simulation output and compare it with the human activity system with the objective of creating a shared understanding of the results of the work practice model and simulation. The

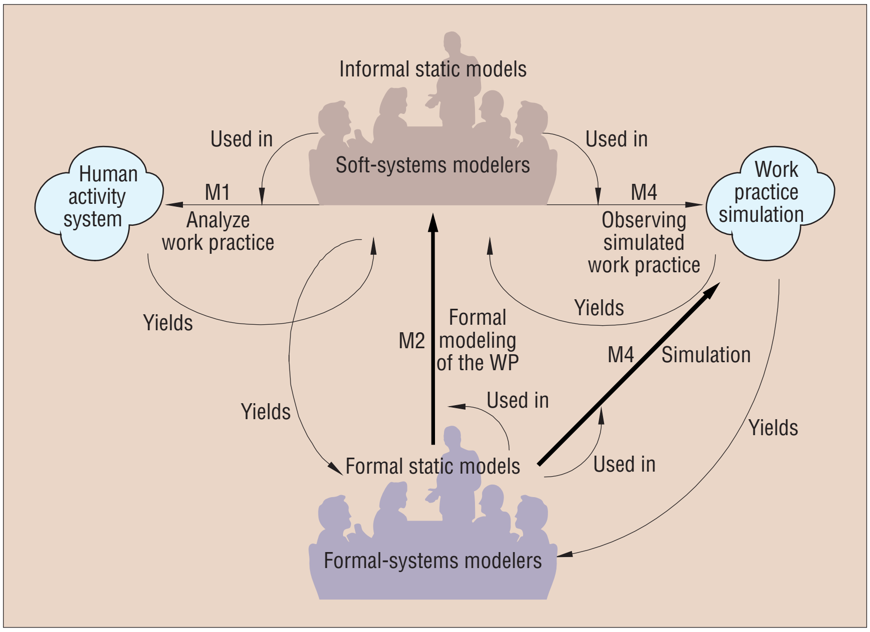

Figure 1. Brahms work practice modeling process of the human activity system.

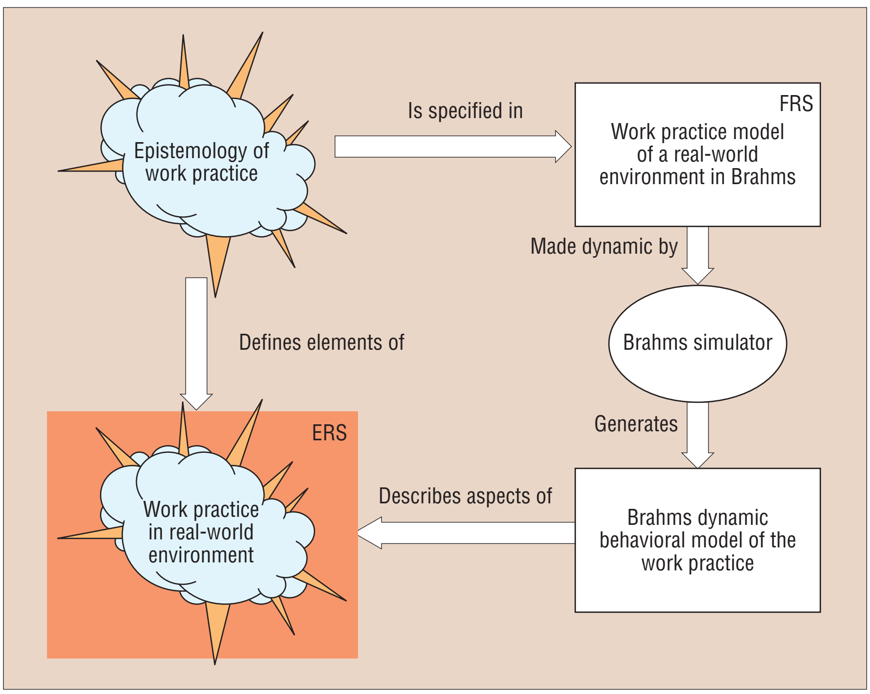

Figure 2. Describing real-world work practice (empirical relational system) with computational modeling (formal relational system).

result might be suggestions for changes to the formal model—for example, to perform a what-if scenario. Thus, there is a modeling and simulation cycle between M1, M2, M3, and M4, which means these methods must be closely integrated if we want this cycle to be complete.

We used the Brahms modeling process in the case study described in this article. Because the work system from the case study did not yet exist, the analysis of the work practice (M1) became an analysis of the work system based on similar previous missions and the proposal for the new mission. During this analysis, the work practice analyst worked with the mission project team members (including the principal investigator, mission scientists, and roboticists) who had practical experience with similar work systems. After this initial work practice analysis, the Brahms modeler developed an initial formal model of the work system (M2) in Brahms. Then the modeler compiled, simulated, and fixed the errors in the formal model until there was a high-level simulation (M3) of the complete work system. Next the Brahms modeler reviewed the model and simulation results with the mission project team members to verify the model and get more detailed information about the work practice (M4). The M2–M3–M4 cycle happened three times over a period of six months. Our case study gives a more detailed explanation of the development of a Brahms model as a part of M2 and M3.

Developing a model of work practice

Figure 2 shows how our epistemology of work practice, formalized in the Brahms modeling language and made operational in the Brahms simulation engine, relates a simulation to a real-world human activity system. The empirical relational system (ERS in Figure 2) is the human activity system under observation in Figure 1. The purpose of Brahms is to make modeling the ERS possible and to create a work practice model that the Brahms simulator can execute.

In general, Brahms models represent work with much more detail than business process models but somewhat less detail (and far more broadly) than cognitive models. Considerable effort is devoted to modeling objects (for example, fax machines) and computer systems, with which people often interact to accomplish their work. The Brahms language and simulation engine relates several levels of detail (areas and objects, groups and agents, activities and actions) and integrates different perspectives—physical, cognitive, and social.

Typically, a modeler sketches a Brahms model by specifying the geography and groups first. The grain size of the simulation clock (time per tick) might vary from one second or less to five minutes or more, depending on the information available and modeling purposes. Common objects and activities such as telephones and phone conversation can be easily reused or adapted, by the modeler, from other Brahms models.

Work practice is a set of related models that we can view independently, which makes the modeling effort easier:

- *Agent.* The agent model is a group-agent membership hierarchy of the people in the work system. Brahms groups can represent formal roles and functions or be based on location, interpersonal relations, interests, and so forth.
- *Object.* The object model is a class hierarchy of all the domain objects and artifacts—for example, tools, desks, documents, and vehicles.
- *Geography.* The geography model describes areas in which agents and objects are located, consisting of area definitions (user-defined types of areas such as buildings, rooms, and habitats) and areas (instances of area definitions).
- *Activity*. The behaviors of agents and objects are expressed in terms of the activities they perform over time.[5] Agent or object activities are mostly represented at the group or class level, but they are also often specific to agents and objects. Activities are inherited and blended through a priority scheme.
- *Timing.* Constraints on when the activities in the activity model can be performed are represented as preconditions of situation-action rules (*workframes*). Activities take time, as determined by the predefined duration of primitive actions. Workframes can be interrupted and resumed, making the actual length of an activity situation-dependent.
- *Knowledge.* An agent's reasoning is represented as forward-chaining production rules (*thoughtframes*) that fall at group and class levels and can be inherited. Inquiry is modeled as a combination of activities (such as detecting information, communicating, and reading or writing documents) and thoughtframes. Perception is modeled as conditions attached to workframes (called *detectables*). Thus, observation depends on what the agent is doing.
- *Communication.* The communication model describes actions by which agents and objects exchange beliefs, including telling someone something or asking a question. A conversation is modeled as an activity with communication actions, either face to face or through some device such as a telephone or email. The choice of device and how it is used are part of the work practice.

Mission operations system design

As an example, let's look at an initial design of a mission operations system for a proposed NASA discovery mission to the Moon with a semiautonomous rover. This case study illustrates the Brahms modeling approach and the potential gain for engaging in such modeling. Specifically, the modeling produced useful insights about power consumption and its potential impact on science objectives under certain scenarios in the Victoria mission.

Victoria is the name of a proposed long-term semiautonomous robotic mission to the south pole region of the Moon. (The mission was named after Ferdinand Magellan's only ship that completed the circumnavigation of the world.) At the start of this case study, the NASA Victoria team was in the middle of writing a mission proposal. The Victoria mission's team members (principal investigator and coinvestigators) are world-renowned scientists from different scientific disciplines: planetary scientists, geologists, roboticists, and artificial intelligence specialists.

Victoria's primary mission is to verify the presence of water ice and other volatiles in permanently shadowed regions on the Moon. This will be accomplished by gathering lunar data for analyzing the history of water and other volatiles on the Moon and, by implication, in the inner solar system. The team decided the most efficient approach would be to use a high-speed semiautonomous rover that could traverse a long distance (several hundreds of kilometers) for a long time period (three months to a year) while gathering the necessary geological and physics data.[11] Using the Brahms approach, we developed a model of the total mission operations work system during the proposal phase of the project, including a model of people's work practice in mission operations, the rover on the Moon, the information systems, and people's workspaces. With Brahms, we were able to quantify the impact of the human work practice on the productivity of the rover on the Moon.

The work system is centered on remote human–robot interaction. On the basis of the rover science data returned, the Earth-based science team decides what rover commands to send next (while trying to maximize the quantity and quality of the returned science data). We should consider the rover as a servant to the science team.

The Victoria mission's work is distributed over several human teams and the Victoria rover. In a sense, we can view the Victoria science team as a user of the teleoperated rover. On the other hand, by virtue of being people's arms and eyes on the Moon, the rover is more of an assistant than a simple tool. In particular, the work is distributed between people and robot, so we can ask, how do the behaviors of people and the robot interact? Who is doing what, where, when, and how?

The Victoria model is limited because agents represent teams. We did not model how decision making within and between teams will occur, making the model less complete. However, our modeling approach is incremental.

Victoria mission operations work system

Figure 3 is an informal representation of the people and objects in the Victoria work system and their locations during the mission. The science team consists of several subteams collocated in Building 244 at the NASA Ames Research Center in California: the science operations team (SOT), the instrument synergy team (IST), and the data analysis and interpretation team (DAIT). Two other supporting teams are outside the science team: the data and downlink team (DDT) and the vehicle and spacecraft operations team (VSOT). These teams work together to accomplish the mission's scientific objectives, which involve acquiring certain data from different locations on the Moon. The teams communicate with the Victoria rover on the lunar surface using the universal space network (USN) via two separate communication links: the high-capacity S-Band direct Earth-to-rover link and the UHF communication link (directly and via Victoria's lunar orbiter).

Telemetry and science data will come to NASA Ames via the universal space network data connection. Data will be automatically converted by mission information systems into accessible formats in near real time and made available to the teams via visualization applications (such as 3D visualization of the lunar surface).

The model's purpose

The major limitation of current robot energy modeling tools, apart from problems with creating and revising models, is their inability to include human factors in their calculations of the rover's power consumption. Before our case study, the impact of Earth-

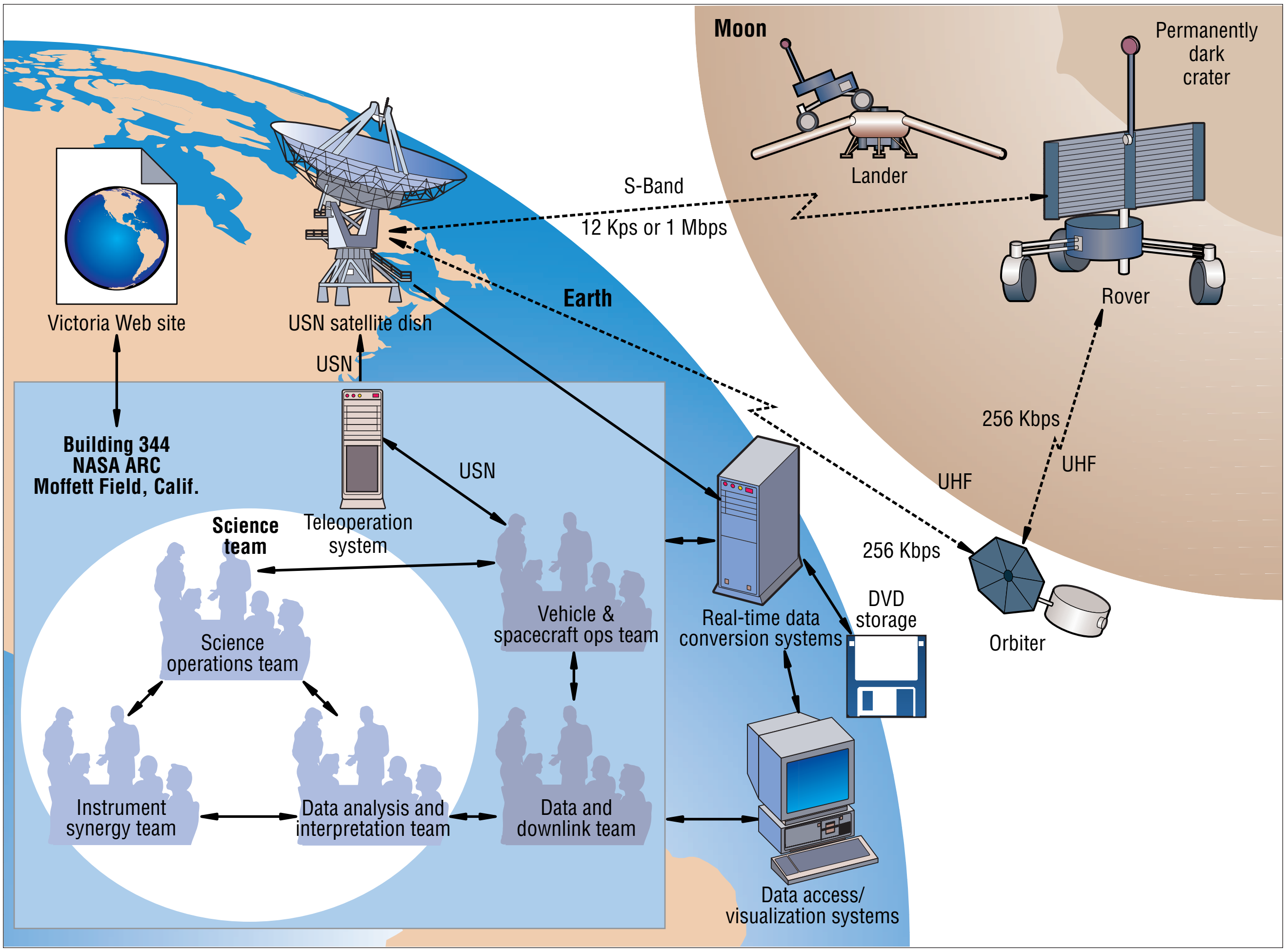

Figure 3. The Victoria lunar mission work system.

based operations on the rover's energy usage was unknown. Consequently, one purpose of the case study was to determine the effect of a particular work system design on the rover's power consumption during a science traverse into a permanently dark crater.

The Brahms–Victoria model prescribes a work system design by modeling the rover and the team's geographical locations and movements; the activities of all the Earth-based teams, the rover, and the communication actions of both; as well as the mission information systems the teams are using. Through this example of the Victoria mission, we are also explicating the Brahms modeling language and how the components interact in work practice simulations.

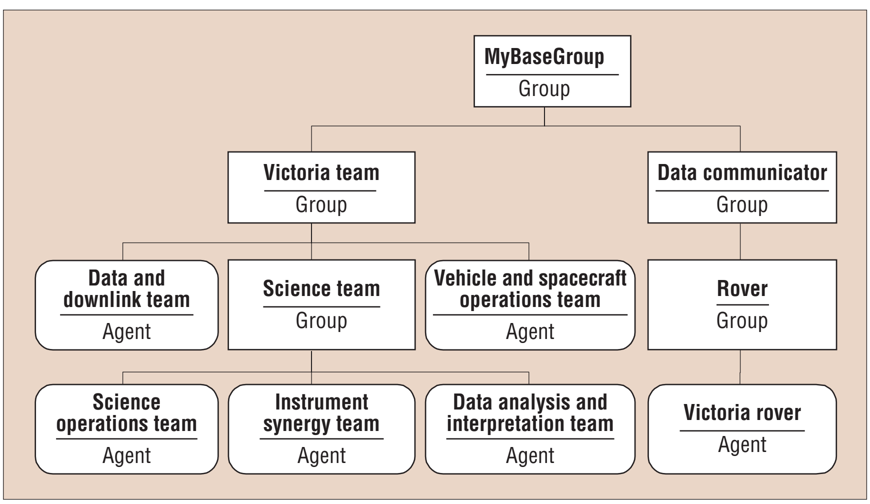

Figure 4. The Victoria agent model.

Agent model

Figure 4 shows the group–agent membership hierarchy on which the work system's design is based. The agents in the model are the Earth-based human teams and the Victoria rover, as shown in Figure 3. The teams are represented as single agents because at this moment prescribing each team's composition and practices in more detail is not possible. For example, the SOT's "plan a command sequence" activity represents the team's work, whereas each team member's individual activities remain unspecified.

Table 1. Functional activity distribution over Victoria teams.

Process	Science operations team	Instrument synergy team	Data analysis and interpretation team	Data and downlink team	Vehicle and spacecraft operations team	Rover
Uplink	1. Maneuver commands 2. Command sequences for experiment operation	1. Commands for engineering operation of robot or spacecraft 2. Emergency or anomaly resolution commands	1. Long-term planning for science opportunities	1. Telecom-munications commands	1. Maneuver commands 2. Command sequences for experiment operation	1. Command execution
Downlink		1. Monitoring of health and status telemetry from robot subsystems	1. Data quality assessment 2. Experiment data collection	1. Experiment data collection 2. Data processing and enhancement		1. Experiment data collection

VictoriaRover is modeled as an agent because it has behaviors (including primitive actions that change the world), movements, and communications. Strictly speaking, activities of designed objects are only formal processes, whereas activities of people are conceptualizations of actual behavior. However, in a Brahms model, both are abstractions in a formal language. So, the distinction is how we interpret the model—what it represents, rather than how the simulation executes the activities of agents versus the activities of objects.

Table 1 shows a possible distribution of the functions over the Victoria teams.[12] Details of how different teams collaborate to perform these functions constitute the work practice of the different agents, specified in Brahms workframes and expressed as situation-action rules.

An example SOT workframe for creating a command sequence for finding water ice is (paraphrased): "When I believe that there is a possibility we can find water ice at the current location of the rover, then start the activity of finding water ice." Generically, a workframe is of the form, "When (I believe {X}*) Do {activity A, conclude a new belief and/or fact}*."

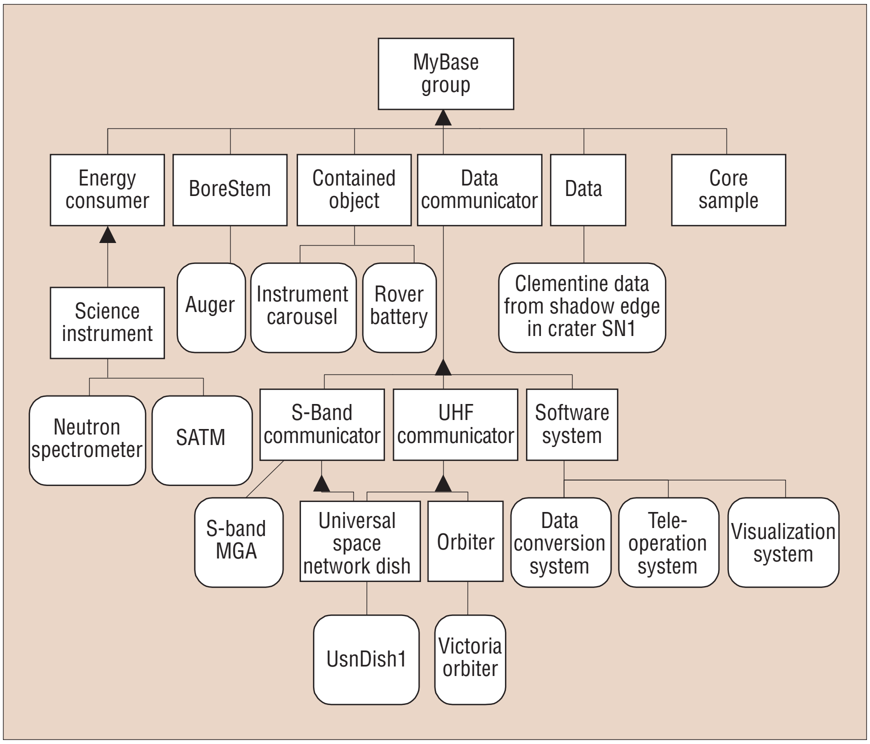

Figure 5. The Victoria mission object model.

Object model

A Brahms object model consists of the classes and instances of physical artifacts as well the data objects created during the simulation. The Victoria object model (see Figure 5) includes classes for the science instruments on the rover as well as other objects contained in the rover, such as the carousel and the battery. The data communicator class includes the objects for S-Band and UHF communication. The model also represents software systems that receive, convert, and visualize mission data. The Data and CoreSample classes dynamically create data instances and lunar core sample objects during the simulation.

Geography model

The Victoria geography model (see Figure 6) represents locations on Earth and the Moon. The dotted lines in Figure 6 show class-instance relationships, whereas the solid lines show part-of relations.

The Victoria teams and systems are located in Building 244 at NASA Ames Research Center, and the UsnDish1 satellite dish is located in the area UsnSatelliteLocation. Locations for the simulated scenario are represented on the Moon; ShadowEdgeOfCraterSN1 represents the rover's location at the start of the simulation (the shadow edge that is in crater site number 1). ShadowArea1InCraterSN1 represents the location in the permanently shadowed SN1 crater where the rover will perform a drilling activity. The LandingSite area is only represented for completeness.

Victoria simulation scenario

The Victoria proposal spells out many surface activities that the rover will perform in coordination with the teams on Earth. For this case study, we selected the activity of

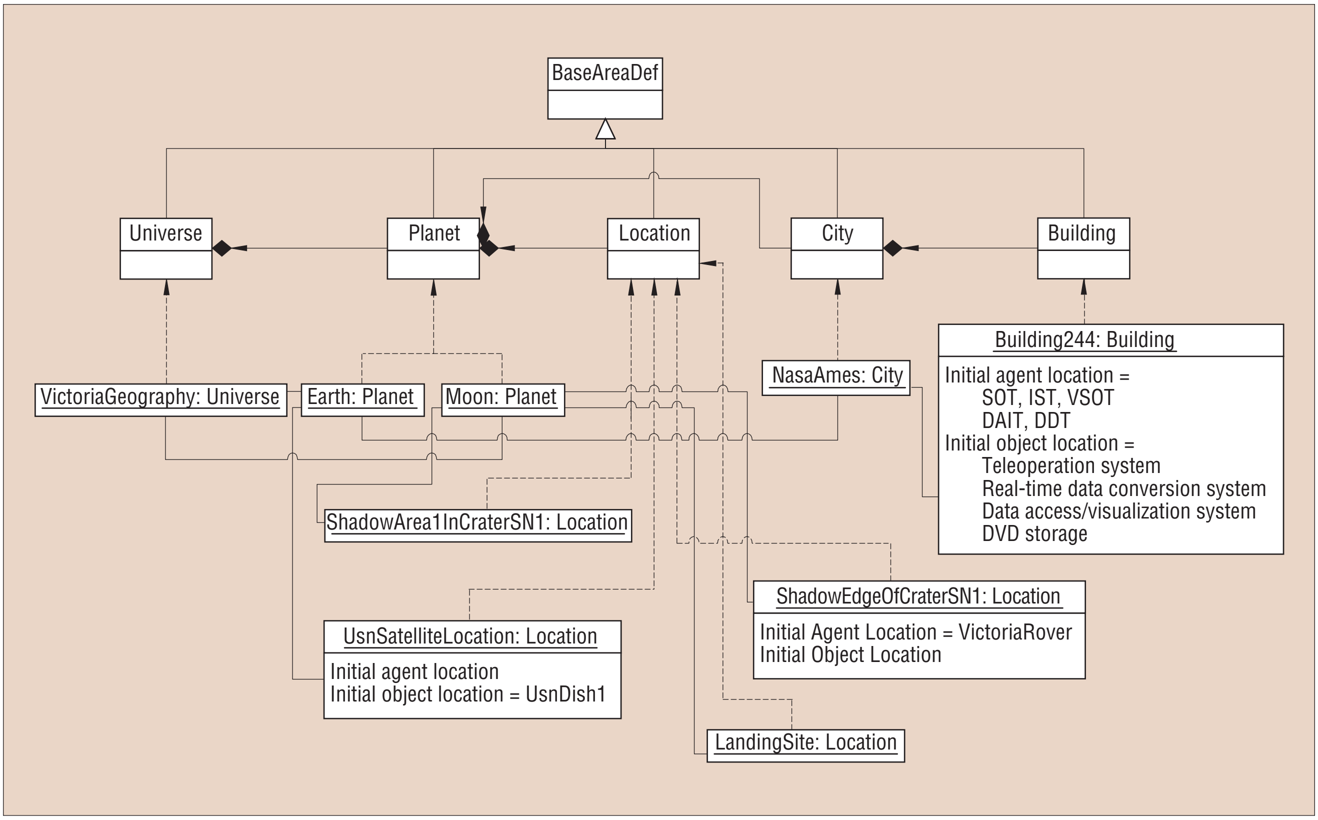

Figure 6. The Victoria geography model.

searching for water in permanently shadowed craters:

> The rover arrives at the shadow edge of crater site number 1. The battery is fully charged. On the basis of the data analysis by the Earth-based teams of the Clementine data available for the shadow edge area of crater site number 1, the science team decides where to enter this crater and search for water ice. As the rover enters the crater, it takes hydrogen measurements with the neutron spectrometer. When the rover arrives at the assigned location within this crater and finds hydrogen there, the science team decides to drill 10 cm into the surface using the sample acquisition and transfer mechanism (SATM) and collect a 1.0-cc lunar sample. When the rover receives this command, it starts the drilling activity and finally deposits the sample into the instrument carousel.

The rover uses two instruments in this scenario: the neutron spectrometer (to detect hydrogen—most likely caused by water ice—within the first half meter of the lunar surface below the rover) and the SATM.

The simulation model's backbone consists of three primary activities: data uplink, rover operations, and downlink.

Simulation results

The simulation lets us visualize the work system's behavior over time—that is, activities, communication, and movement of each agent and object in the work system. The Brahms simulation engine executes the model after it is compiled. The simulation engine creates a relational database, including every simulation event. A Brahms model display tool called the AgentViewer uses this database to display all groups, classes, agents, objects, and areas in a selectable hierarchical browser. The AgentViewer's user can select the agents and objects he or she wants to investigate to understand what occurred during the simulation. The AgentViewer displays an activity timeline of the selecting agents and objects, highlighting the communications. Let's look at the results of the Victoria model simulation based on screenshots from the AgentViewer application.

Data uplink

The scenario starts with the DAIT retrieving the Clementine data image of the shadow edge area, where the rover is located. The team reviews this image using the visualization system, represented by the VisualizationSystem object.

According to the work practice, the DAIT does this without anyone requesting to look at the data. This means that it needs to know

- The rover's location and situation
- Whether data is available and needs to be retrieved
- Where and how it can retrieve data

Once the DAIT has retrieved the images, it communicates this to the SOT, and the two teams collaboratively analyze these images (the AnalyzeRoverImages activity). At the end of this analysis activity, the SOT plans the first rover command sequence. According to the scenario being simulated, the SOT decides the rover needs to drive for a specified time (15 minutes) into the crater to a specific location (ShadowArea1InCraterSN1) and that while driving it should use its neutron detector instrument to detect hydrogen in the lunar surface. This decision is communicated to the VSOT (and the DAIT). After this communication, the SOT waits for the rover's downlink data.

Rover

The Victoria rover is modeled as an agent, whereas the neutron spectrometer and SATM instruments are modeled as separate science instrument objects contained in the rover. In the scenario, the NeutronSpectrometer object is active and creates a HydrogenData_1 object containing the hydrogen data that are sent to Earth while the VictoriaRover is traversing to a permanently shadowed area within the crater SN1 (see Figure 7). The rover then waits for the next command sequence from Earth. Meanwhile, the Earth-based teams analyze the hydrogen data and decide what to do next. In the second uplink activity, the VSOT commands the rover to search for water ice in the permanent dark area. This triggers the SATM instrument to start the drilling activity.

To collect a sample the SATM must

- Lower its auger to the surface
- Drill to the depth given as part of the command by the Earth-based science team (in this scenario, the command is to take a 1.0-cc sample at a 10-cm depth)
- Open the sample cavity door
- Continue drilling to collect the sample
- Close the sample door when done
- Retract the drill from the surface
- Deposit the collected sample on the instrument carousel (see Figure 7)

The activity durations for drilling into the surface are dynamically derived during the simulation of the rover's drilling activities. Honeybee Robotics, the designers and manufacturers of the SATM instrument, provided data for the time it takes to move the auger to the surface and to open and close the sample door, and the average time it takes to drill the auger into, and retract it out of, the lunar surface.

Downlink

When the rover detects hydrogen in the ShadowArea1InCraterSN1 location, the downlink process starts. The Brahms AgentViewer in Figure 8 demonstrates this.

The VictoriaRover agent contains the S-Band medium-gain antenna object, which represents the S-Band transmitter on the rover. The VictoriaRover creates a data object with both the current rover location information and the hydrogen data. This data object is then communicated to Earth via the UsnDish1 object. The UsnDish1 object communicates the data to the DataConversionSystem located at NASA Ames. As Figure 8 shows, the DataConversionSystem performs two conversion activities, one for the hydrogen data and one for the location data from the rover. The work system design requires the data conversion system to interact with the visualization system without human intervention. Requirements for these systems could have easily been modeled in more detail in the Brahms model, but this was not the focus of the case study.

Figure 7. Victoria rover scenario activities.

When the VisualizationSystem receives the newly converted data, the system alerts the DAIT. A member of the DAIT monitors the VisualizationSystem while in the activity WatchForDownlink. When the DAIT agent detects newly available neutron detector and location data, it retrieves the data from the VisualizationSystem object (that is, the activities RetrieveNeutronData, InterpretNeutronData, and FindRoverLocationData). This simulates the DAIT members looking at and interpreting the rover's neutron and location data using the visualization system.

Next, the DAIT communicates its findings to the SOT. In this example scenario, the hydrogen data suggest that the rover has found hydrogen in the ShadowArea1InCraterSn1 area. Given this finding, the SOT determines the next command sequence for the rover and communicates this decision to the VSOT.

The communication informs the VSOT to transmit the command sequence to the VictoriaRover. The command sequence tells the VictoriaRover to start the SearchForWaterIceInPermanentDarkArea activity. It also tells the VictoriaRover that its subactivity, which should occur during this primary activity, is to perform the DrillingActivity. Parameters indicate how deep to drill and how big a sample to collect at that depth. Figure 8 shows part of this second uplink process.

The duration of this downlink and second uplink process determines the length of the VictoriaRover's DoNothing activity, representing the time the rover must wait for the Victoria science team to design the next command sequence.

Modeling the rover's energy consumption

The scenario identifies the energy usage during all the rover's primitive activities, based on each subsystem and instrument on the rover requiring power during a specific activity. The rover designers, the Robotics Institute at Carnegie Mellon University, provided the power consumption specifica-

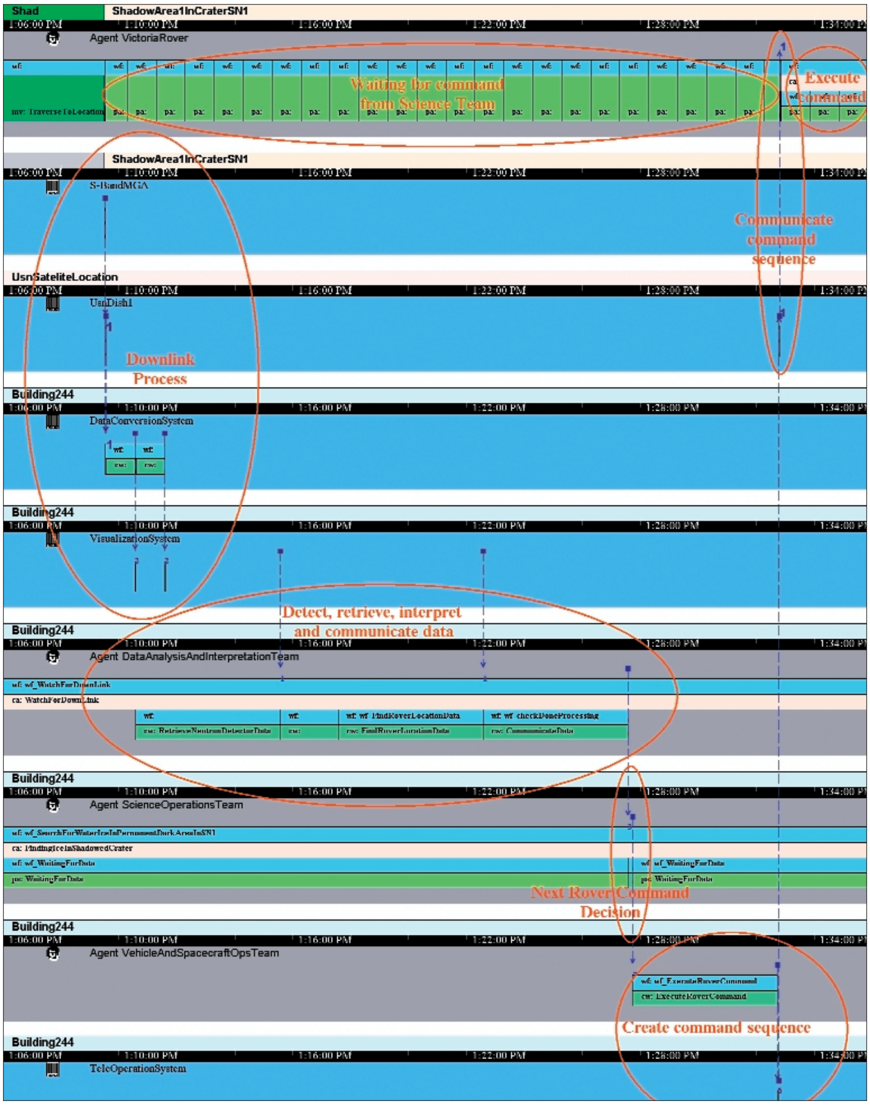

Figure 8. Simulation of downlink and second uplink command activities.

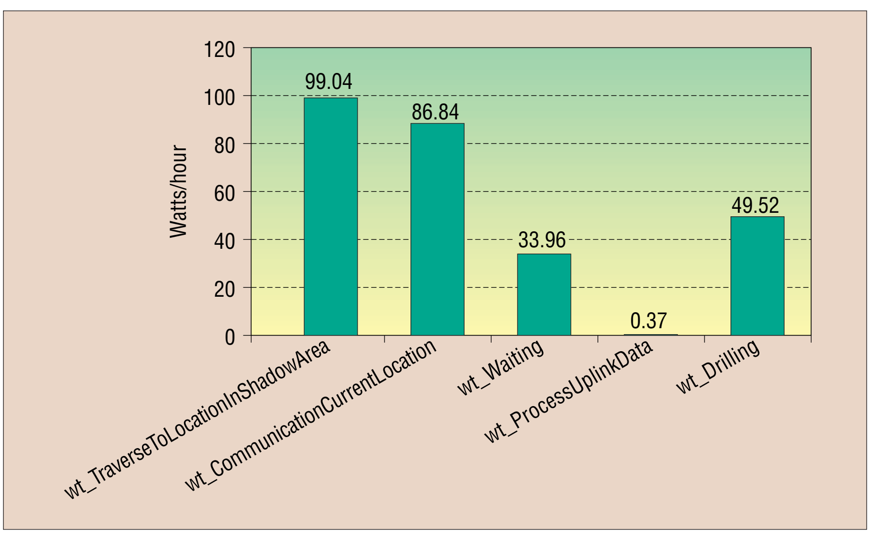

Figure 9. Rover energy used in high-level activities from simulation history database.

tion for the rover's low-level activities. Using Equation 1, the simulation calculates energy usage during each rover activity in Equation 2:

Power consumption for rover at time t ($Prover(t)$)
= power for driving
+ power for command & data handling
+ power for science instrumentation
+ power for communications
+ power used for thermal protection
+ other (not measurable power) (1)

$$Eact_i = \int_{start\ of\ act_i}^{end\ of\ each\ act_i} Prover(t)\,dt \quad (2)$$

The rover's total power consumption during the scenario can then be calculated, in the simulation, by adding all the energy usages for each rover activity:

$$\text{Total Power Consumption} = \sum_{i=0}^{n} Eact_i \quad (3)$$

Figure 9 shows energy consumption for every rover activity during the simulation. The energy the rover uses during the waiting activity is defined by the energy needed for *Thermal Protection during driving + Command and Data Handling during driving.* This means that even while the rover is standing still and "doing nothing," it consumes power for its thermal protection and its command and data handling for its subsystems.

Another interesting variable is the rover's energy usage rate. The power level in the battery object at the start of the scenario is in watts/hour. The simulation calculates how much power the rover uses, in watts, on the basis of the activities it performs over time, resulting in a total power consumption at the end of the scenario. The EnergyRate is the percentage of power usage of the rover for the scenario: the amount of power consumed by the rover in this scenario given the total power available in the battery at the beginning of the scenario.

EnergyRate
= *Total Power Consumption* (W/hr)/
Power of battery at start of scenario (W) (4)

Recalling the scenario involves driving 900 meters into the crater and taking one 1.0-cc sample at 10-cm depth, and using Equations 3 and 4 (together with the data corresponding to the current work system design), we find the EnergyRate equals approximately 0.30 (or 30 percent per hour). In other words, the rover consumes almost one-third of its power during the scenario. The energy consumption rate of the rover was higher than the Victoria team expected, because the time the rover had to wait in the permanent dark crater for the next command from the science team (see Figures 6, 7, and 8) had been previously left out of the mission design. While waiting, the rover consumes thermal and communication power. The longer the downlink–uplink decision cycle of the mission operation teams, the more power the rover consumes.

The length of this human decision cycle is dependent on the work system design; thus, the energy consumption rate of the rover (EnergyRate) is also dependent on the work system design. This variable represents the work system design's rover power efficiency and is a measure that other researchers can use to compare different work system designs for their chosen scenarios.

We believe that the Brahms language approach and simulation engine are still in their infancy, with decades of research and application required before we have accomplished our ultimate objective of usefully modeling the complexities of human behavior in work settings. In viewing work broadly as part of human life, many possible aspects might be relevant: the nature of identity as played out in interpersonal interactions (for example, office politics and friendships), anthropometric details (such as the ability to reach controls), decision making (cognitive models of reasoning), fatigue, boredom, diurnal rhythm, "external life" (errands and family interruptions), and learning (especially by watching and mimicking). We require further research and experience in using Brahms in design projects to decide which of these perspectives to include and at what level of detail.

Practical challenges to developing reusable model components organized by types of settings and human interactions exist. To use Brahms to explore a variety of workload conditions, it would be useful to have tools for statistically generating cases for simulation analysis. We would also need theoretical frameworks for validating analog models (for example, relating Arctic expeditions to space station experience and planned missions to Mars).

Each model we construct is both an experiment and a revelation. Every setting changes our understanding of work practice and the requirements for modeling it. The practical boundaries of what is necessary for work systems design and what is only of research interest remain to be seen.

References

1. C.H. Pava, *Managing New Office Technology: An Organizational Strategy*, Free Press, New York, 1984.
2. R.J. Mayer et al., "Framework and a Suite of Methods for Business Process Reengineering," *Business Process Change: Reengineering Concepts, Methods and Technologies*, V. Grover and W.J. Kettinger, eds., Idea Group, Hershey, Pa., 1998.
3. P. Ehn, *Work-Oriented Design of Computer Artifacts*, Arbetslivcentrum, Stockholm, 1988.
4. J. Greenbaum and M. Kyng, eds., *Design at Work: Cooperative Design of Computer Systems*, Lawrence Erlbaum Associates, Mahwah, N.J., 1991.
5. W.J. Clancey, "Simulating Activities: Relating Motives, Deliberation, and Attentive Coordination," to be published in *Cognitive Systems Rev.*, 2002.
6. W.J. Clancey et al., "Brahms: Simulating Practice for Work Systems Design," *Int'l J. Human-Computer Studies*, vol. 49, 1998, pp. 831–865.
7. M. Sierhuis, *Modeling and Simulating Work Practice. Brahms: A Multiagent Modeling and Simulation Language for Work System Analysis and Design*, SIKS Dissertation Series no. 2001-10, Dept. of Social Science Informatics, Univ. of Amsterdam, Amsterdam, 2001.
8. E. Yourdon, *Modern Structured Analysis*, Prentice-Hall, Upper Saddle River, N.J., 1989.
9. K Holtzblatt and H. Beyer, *Contextual Design: A Customer-Centered Approach to Systems Design*, Morgan Kaufmann, San Francisco, 1998.
10. J. Blomberg et al., "Ethnographic Field Methods and Their Relation to Design," *Participatory Design: Principles and Practices*, A.N.D. Schuller, ed., Lawrence Erlbaum Associates, Mahwah, N.J., 1993, pp. 1, 26, 123–155.
11. N.A. Cabrol et al., "Science Results of the Atacama Nomad Rover Field Experiment, Chile: Implications for Planetary Exploration," *J. Geophysical Research*, vol. 106, no. E4, 2001, pp. 7664–7675.
12. S.D. Wall and K.W. Ledbetter, *Design of Mission Operations Systems for Scientific Remote Sensing*, Taylor & Francis, London, 1991.

The Authors

Maarten Sierhuis is a senior research scientist at the Research Institute for Advanced Computer Science (an institute of the Universities Space Research Association at NASA Ames Research Center), where he manages the Brahms project on modeling and simulating work practice. His research interests are multiagent modeling languages and their application to the development of human-centered systems. Before joining RIACS, he was a member of the Work Systems Design group and the Expert Systems laboratory of NYNEX Science & Technology. He also developed expert systems as a senior knowledge engineer in the Netherlands and at IBM. He received his PhD from the Department of Social Science Informatics at the University of Amsterdam. Contact him at RIACS, M/S T35B-1, NASA Ames Research Center, Moffett Field, CA 94035-1000; msierhuis@mail.arc.nasa.gov.

William J. Clancey is a senior research scientist at the Institute for Human & Machine Cognition, University of West Florida, Pensacola, and chief scientist for human-centered computing at NASA Ames Research Center, Computational Sciences Division. His current interest is the relation of descriptive cognitive theories to human experience and neural processes. Before joining IHMC and NASA, he was a founding member of the Institute for Research on Learning, where he codeveloped the work system design methods of business anthropology in corporate environments. He also did research in artificial intelligence at Stanford University's Knowledge Systems Laboratory. He received his PhD in computer science from Stanford University. He is a member of the steering committee of the Mars Society and serves as a NASA Visiting Researcher for the Challenger Center's school outreach program. Contact him at the Computational Sciences Division, M/S 269-3, NASA Ames Research Center, Moffett Field, CA 94035; bclancey@mail.arc.nasa.gov.