Multiagent Work Practice Simulation: Progress and Challenges

William J. Clancey[1], PhD and Maarten Sierhuis[2], PhD
NASA/Ames Research Center
Computational Sciences Division, M/S 269-3
Moffett Field, CA 94035, USA

Phone 1-650-604-2526, 4-4026 fax
bclancey@arc.nasa.gov, msierhuis@arc.nasa.gov

Summary

Modeling and simulating complex human-system interactions requires going beyond formal procedures and information flows to analyze how people interact with each other. Such *work practices* include conversations, modes of communication, informal assistance, impromptu meetings, workarounds, and so on. To make these social processes visible, we have developed a multiagent simulation tool, called Brahms, for modeling the *activities* of people belonging to multiple groups, situated in a physical environment (geographic regions, buildings, transport vehicles, etc.) consisting of tools, documents, and computer systems. We are finding many useful applications of Brahms for system requirements analysis, instruction, implementing software agents, and as a workbench for relating cognitive and social theories of human behavior. Many challenges remain for representing work practices, including modeling: memory over multiple days, scheduled activities combining physical objects, groups, and locations on a timeline (such as a Space Shuttle mission), habitat vehicles with trajectories (such as the Shuttle), agent movement in 3d space (e.g., inside the International Space Station), agent posture and line of sight, coupled movements (such as carrying objects), and learning (mimicry, forming habits, detecting repetition, etc.).

Background: Brahms and Work Practice Modeling

A Brahms model of work practice (Clancey, et al., 1998) reveals *circumstantial, interactional influences* on how work actually gets done, especially how people informally involve each other in their work, thus changing the quality of the result. In particular, a model of practice reveals how collaboration is accomplished in communications, including meetings, email, workflow systems, and written documents (Wenger, 1998). Choices of what and how to communicate are dependent upon *social beliefs and behaviors*—what people know about each other's activities, intentions, and capabilities and their understanding of the norms of the group. As a result, building a Brahms model leads human-computer system designers to question *how tasks and information actually flow* between people and machines, what work is required to synchronize individual contributions, and how tools hinder or help this process (Greenbaum & Kyng, 1991; Bagnara, 1995). In particular, workflow diagrams generated by Brahms are *the emergent product of local interactions between agents and representational artifacts*, not pre-ordained, end-to-end paths built in by a modeler.

To illuminate how formal flow descriptions relate to the social systems of work, Brahms incorporates multiple views—relating people, information, systems, and geography—in one tool. Such views help work system designers, managers, and trainers better understand the interactive, circumstantial

---

[1] On leave from the Institute for Human and Machine Cognition, University of West Florida, Pensacola
[2] Employed by the Research Institute for Advanced Computer Science (RIACS).

Paper presented at the RTO NMSG Conference on "Future Modelling and Simulation Challenges", held in Breda, Netherlands, 12-14 November 2001, and published in RTO-MP-073.

importance of proximity of people and tools to each other, timing of individual interactions, and how attention is conceptually scoped by work settings and roles. Accordingly, we begin to see how work flow is an abstraction; actual work is accomplished and practices learned through often chance interactions, which are omitted from most process models and written procedures.

Brahms was originally developed as a research tool at a telecommunications company (NYNEX) and the Institute for Research on Learning. More recently, Brahms is being applied at NASA for crew scheduling, human-robot system design, and operations assistants in extreme environments. An example is presented in subsequent sections to illustrate the components and operation of Brahms simulations. Many challenges remain for representing work practices, which we discuss at some length in the last part of this paper.

Basic Components of a Brahms Simulation

A Brahms simulation of work practice has seven components:

Agent Model: The group-agent membership hierarchy of the people in the work system. Groups may be formal roles and functions or based on location, interpersonal relations, interests, etc.

Object Model: The class-hierarchy of all the domain objects and artifacts, e.g., tools, desks, documents, vehicles.

Geography Model: The geographical 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 Model: The behavior of agents and objects in terms of the activities they perform over time (Clancey 1997). Agent or object activities are mostly represented at the group-level or class-level respectively, but are also often specific to agents and objects. Activities are inherited and blended through a priority scheme.

Timing Model: Constraints on when the activities in the activity model can be performed, represented as preconditions of situation-action rules (called *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 Model: An agent's reasoning, represented as forward-chaining production rules (called *thoughtframes*). Thoughtframes can be represented at group/class levels and inherited. Thoughtframes take no time. Inquiry is modeled as a combination of activities (e.g., detecting information, communicating, and reading/writing documents) and thoughtframes. Perception is modeled as conditions attached to workframes (called *detectables*); thus observation is dependent on what the agent is doing.

Communication Model: 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.

Typically a Brahms model is sketched by specifying the geography and groups first. The grainsize of the simulation clock (time per tick) may vary from 5 seconds or less to 5 minutes or more, depending on the information available and modeling purposes. A model might represent a group of people as a single agent, a useful heuristic in redesigning a work system. Common objects and activities such as telephones and “phone conversation” may be easily reused and adapted from other Brahms models. 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 (e.g., fax machines) and computer systems, with which people interact to accomplish their work.

Comparison to Other Process Modeling Methods

Traditional human factors approaches tend to start with specifications or machinery and study the deficiencies in human behavior (i.e., “performance”) with respect to the predefined requirements of the task or systems to be operated. This approach tends to focus on developing tools (such as tests) to predict how people will perform and then developing training to improve human performance.

A complementary approach is possible. One can start instead with a “bottom up” study of people in their work setting and study how they interact to accomplish their goals, including communication, learning, and work arounds. The emphasis is not on human failures, but on success: How do people succeed despite the deficiencies of their tools and given the inherent conflicts and ambiguities in the work situation? The emphasis of this approach, which we call “work systems design” is on improving the tools, procedures, and facilities. Can we invent new ways of using computers, for example, which better fit human preferences and ways of learning, rather than fitting people to given procedures and tools? Rather than just changing the interface, can we reconceive how the work is done? This same perspective, which focuses on *deficiencies of machines* relative to human capabilities, is essential for developing better “intelligent” computer tools.

Brahms is a simulation tool for representing the interactive behaviors of people and objects in a simulated world. The focus is on how people, tools, and the environment influence each other, such that a total system can be understood and improved. Perhaps the best way to describe Brahms is to contrast its architecture, model content, and how models are developed with other modeling tools:

- Architecture
  - Components are modular and reusable (groups, agents, locations, objects, etc.).
  - Brahms models behaviors, not just inferences; work product flows are output from model, not specified.
  - Behaviors are activated via subsumption (parallel activation; not a procedure stack, activities are not functions or tasks, but how people conceptually organize their time, e.g., relaxing in the evening).
  - Attention (perception of the world) is scoped by activity; i.e., what an agent notices depends on what he/she is doing.
- Content
  - Environment is modeled explicitly, including movement of people through offices, rooms, buildings, and geographic locations (e.g., space station modules).
  - Environment, objects, and agent behaviors interact (not just describing work flow or reasoning).
  - Models represent more detailed causal relations than in conventional process models, indicating how connectivity happens (how processes flow, not just drawing lines between boxes or specifying mathematical relations).
  - Primary focus is on whose knowledge is called into play (participation influences work quality) not what idealized knowledge is required to perform a task.
- Development and Use
  - Ethnography (observing as a participant in the work setting) is primary source of data.
  - Video analysis (of everyday work setting) is essential source of data.
  - Participatory design (including people being studied in the design team) provides primary context for developing and using models.

In effect, Brahms derives from the sociotechnical systems approach of the 1950s (e.g., see Corbet, Rasmussen, and Rauner, 1991), realized in object-oriented computer simulations that combine the methods of qualitative modeling ("artificial intelligence"), cognitive modeling ("knowledge-based systems"), and interactive rendered displays ("virtual reality" and "web-based browsers"). Perhaps most important, Brahms modeling involves a thorough collaboration between social and computer scientists, so interpersonal relations and information processing perspectives are related throughout the study and design process.

Since the initial design of Brahms in the early 1990s, other "multiagent" modeling systems have been developed (see Clancey et al., 1998 for references). No single system is superior for all applications, but we can describe some of the advantages of Brahms relative to other advanced technologies:

- Architecture
  - Agents (and objects) are both deliberative (actions derive from inferences using models of behavior and the environment) and reactive to the environment (actions are immediate and associational).
  - Agent beliefs are independent of facts representing the state of the world.
  - Conceptual objects (e.g., "job orders") allow tracking and abstracting actions (e.g., for determining total time and cost associated with particular work products such as customer orders).
  - Java interface ("API") facilitates integrating other simulations.
- Content
  - Represent communication between agents and objects, plus the communication tools used in specific situations (e.g., fax, phone, email, pager).

Example Application: Victoria Proposed Lunar Mission

To introduce the components of Brahms' language and the nature of the models that can be constructed, we describe a model of a mission operations for Victoria, a proposed long-term semi-autonomous robotic mission to the South Pole region of the Moon. The primary mission objective is to verify the presence of water ice and other volatiles within permanently shadowed regions (Cabrol, et al, in press). During such a traverse the rover will use its neutron detector instrument to detect hydrogen and the Sample Acquisition and Transfer Mechanism (SATM) to drill into the lunar surface and take surface samples to be investigated using an array of science instruments. The essential problem is that the robot needs to have enough power to make it safely out of the dark region before its battery is empty. This makes power consumption a very important constraint in the design of the robot.

MISSION OPERATIONS SYSTEM DESIGN

The work during the Victoria mission will be distributed over a number of human teams and the Victoria rover. By virtue of being people's arms and eyes on the Moon, the teleoperated rover is more of an *assistant* than a simple tool.

Figure.1 represents the work system elements and their relative location during the Victoria mission. The Science Team consists of co-located sub-teams: the Science Operations Team (SOT), the Instrument Synergy Team (IST), and the Data Analysis and Interpretation Team (DAIT). There are two other supporting teams: The Data and Downlink Team (DDT) and the Vehicle and Spacecraft Operations Team (VSOT). The teams communicate with the Victoria rover on the lunar surface using the Universal Space Network (USN), directly and via a lunar orbiter.

The data from the rover will consist mainly of contextual and multi-spectral image data, but will also include thermal emission, a variety of spectrometer data, and microscopic imaging. This data will be automatically converted in near real-time to accessible formats made available to the teams via data visualization applications.

Based on previous experience, the designers hypothesized that the decision cycle of the science team will be affected by many issues, one of which is data overload. They therefore specifically addressed the following questions in the work system design for Victoria:

1. How will science data be gathered collaboratively with the Earth-based science team, rover teleoperator, and the rover on the lunar surface?
2. How will science data be made available to the science team?
3. What is the affect of a particular work system design on the power consumption of the rover during a science traverse into a permanent dark crater?

To answer these questions, Sierhuis (2001) and others developed a model of the activities of the teams, based on the description of a planned mission traverse. In the next sections we describe the design of this work system through the design of the agent model, the object model, their activity models, and the geographical model.

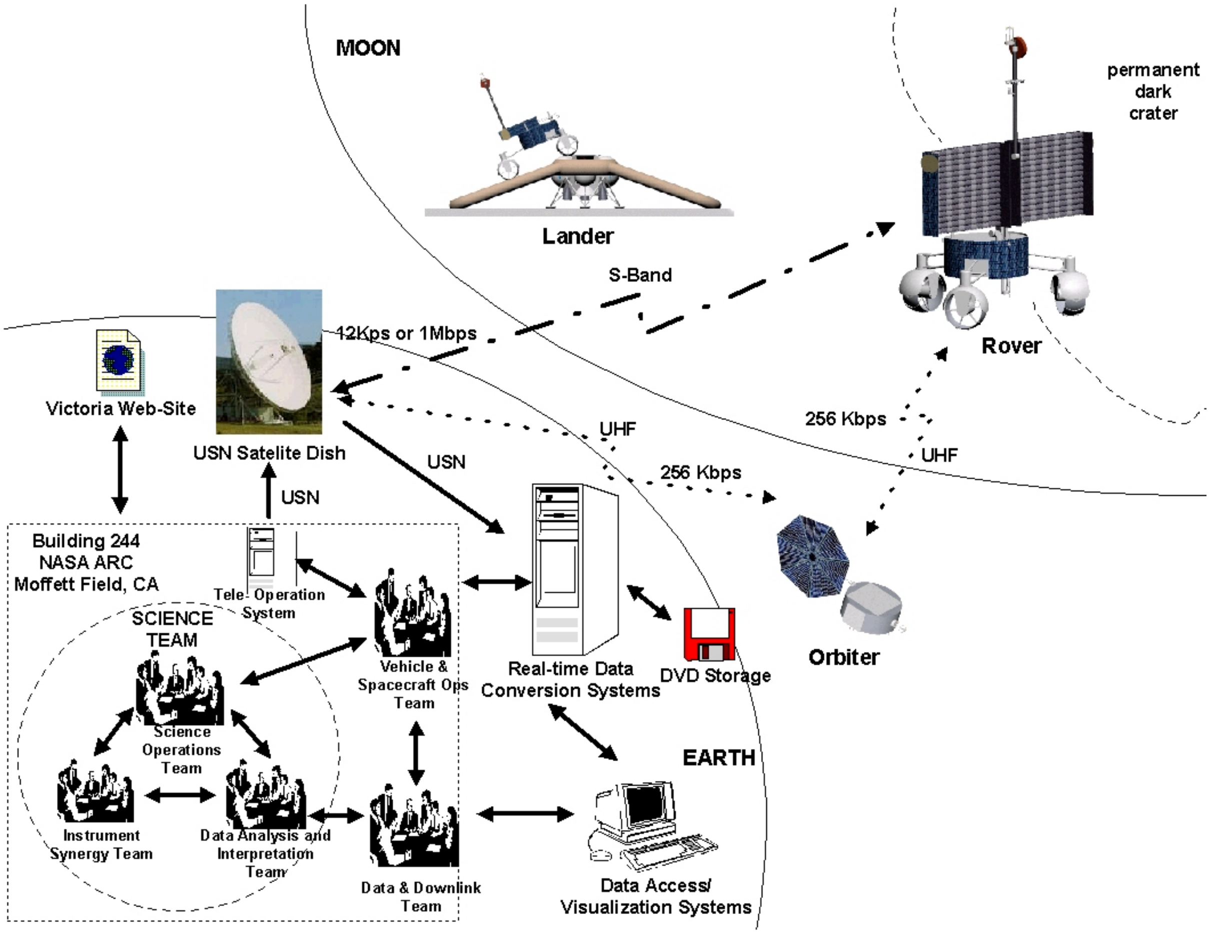

Figure.1. Victoria work system

AGENT MODEL DESIGN

Figure 2 shows the group membership hierarchy on which the design of the work system is based. The agents in the model are the Earth-based human teams and the Victoria rover, as shown in Figure.1. The teams are represented as agents, because it is not yet possible to prescribe the composition and practices of each team in more detail. For example, the "plan a command sequence" activity of the SOT represents the work of the whole team, while the individual activities of each team member remain unspecified. The Victoria Rover is modeled as an agent because it has activities, including primitive actions that change the world, movements, and communications.

Table 1 shows a possible distribution of mission functions over the Victoria teams (Wall, 1991). Details of how different teams collaborate to perform these functions constitute the work practice, as specified in the situation-action rules (Brahms *workframes*) of the different agents. An example workframe for an SOT agent 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}*.*

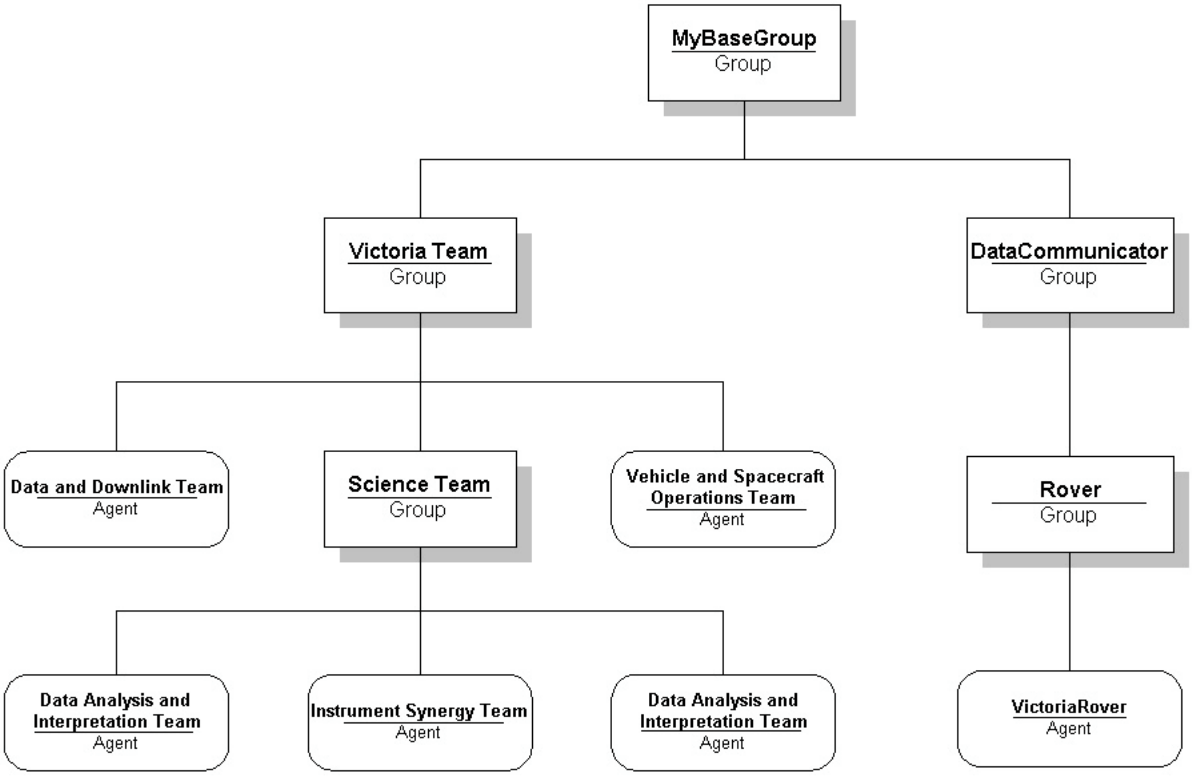

Figure 2. Victoria Agent Model

Object Model Design

The object model consists of the classes and instances of physical artifacts, as well the statically and dynamically created data objects during the simulation. The Victoria object model (Figure 3) includes classes for the science instruments on the rover and other objects contained in the rover, such as the carousel and the battery. Furthermore, the model includes the data communicator class, which includes the objects for S-band and UHF communication. The model also includes the software systems that receive and convert the mission data. A Brahms object represents the data visualization systems that present data to the Victoria team. The Data and CoreSample classes allow dynamically creating objects representing specific data and lunar core samples during the simulation.

GEOGRAPHY MODEL DESIGN

The geography model represents locations on Earth and the Moon (Figure 4). The areas of interest on Earth are Building244, where the Victoria teams and systems are located, and UsnSatelliteLocation, where the UsnDish1 satellite dish is located. Locations for the simulated scenario are represented on the Moon. ShadowEdgeOfCraterSN1 represents the location of the rover at the start of the simulation (the shadow edge in crater SN1). ShadowArea1InCraterSN1 represents the area in the permanent shadowed SN1 crater where the rover will perform a drilling activity. The LandingSite area is represented only for completeness.

Table 1. Functional activity distribution over Victoria teams & Rover

	Science Operations Team	Instrument Synergy Team	Data Analysis and Interpretation Team	Data and Downlink Team	Vehicle and Spacecraft Operations Team	Rover
Uplink process	Maneuver commands Command sequences for experiment operation	Commands for engineering operation of robot/ spacecraft Emergency or anomaly resolution commands	Long-term planning for science opportunities	Telecommunication commands	Maneuver commands Command sequences for experiment operation	Command execution
Downlink process		Monitoring of health and status telemetry from robot subsystems	Data quality assessment Experiment data collection	Experiment data collection Data processing and enhancement		Experiment data collection

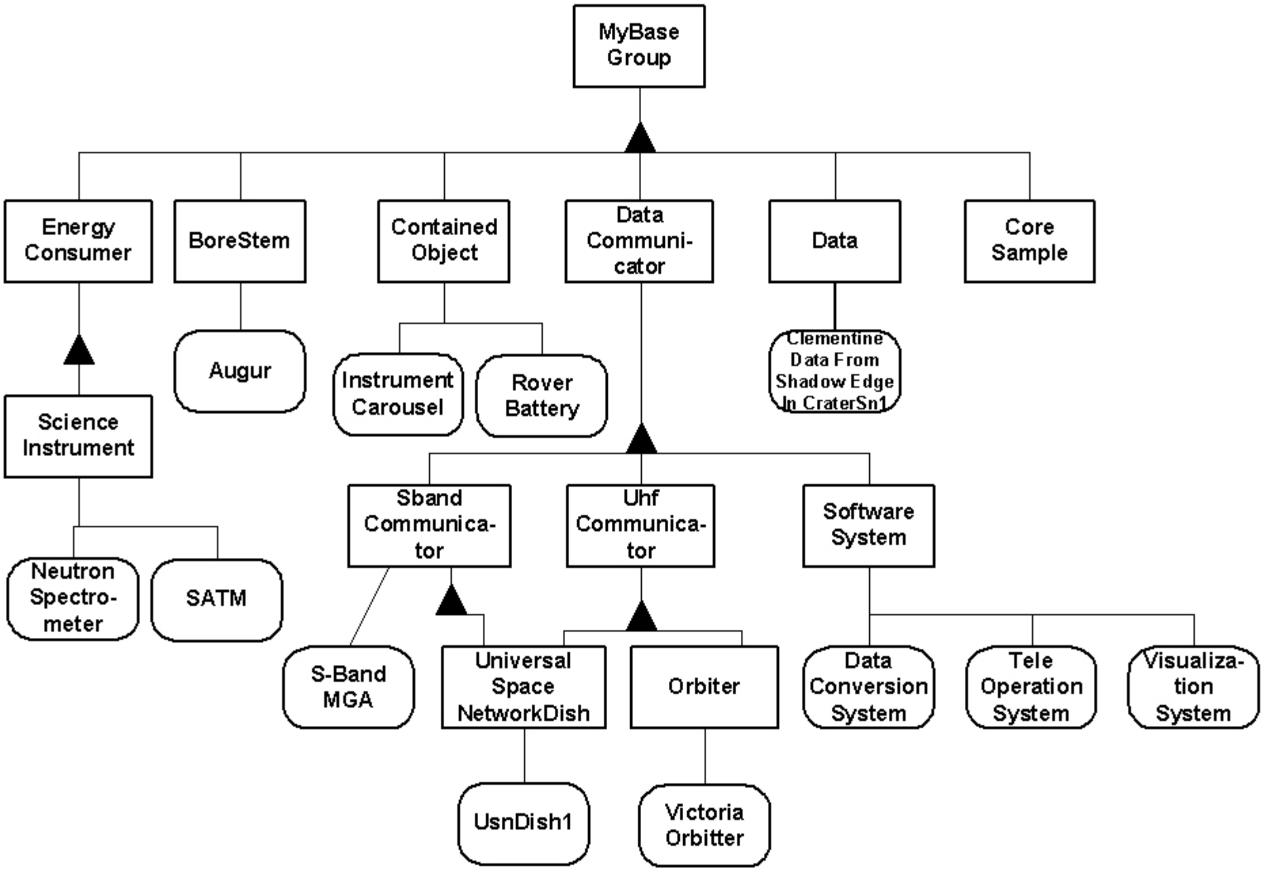

Figure 3. Victoria Object Model

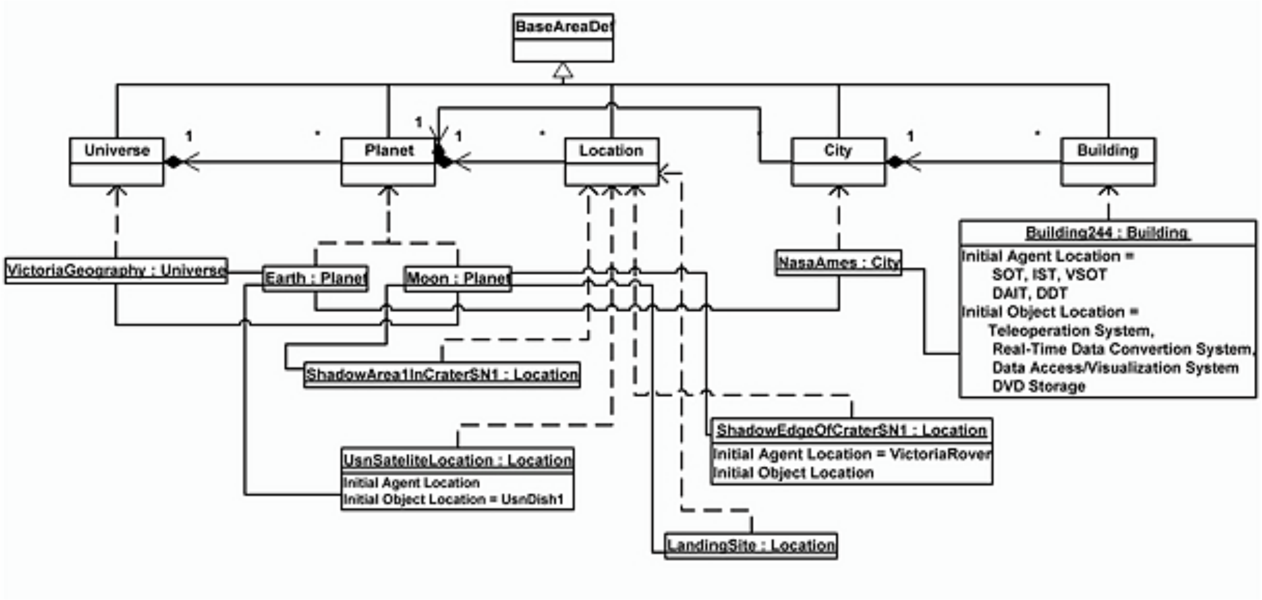

Figure 4. Victoria Geography Model

VICTORIA SIMULATION SCENARIO

The case study selects one of the key surface activities, *searching for water in permanently shadowed craters*:

> The rover has arrived at the shadow edge of crater site number 1. The battery has been fully charged. Based on 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 now decides where to go into this crater and search for water ice. While the rover is traversing into the crater, it is taking hydrogen measurements with the Neutron Spectrometer. When the rover arrives at the assigned location within this crater and it finds hydrogen there, the science team decides it should start drilling 10cm into the surface using the SATM, and collect a 1.0cc 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 lunar surface drill (Sample Acquisition and Transfer Mechanism —SATM).

The backbone of the simulation model consists of three primary activities: Data uplink, Rover operations, and Uplink.

Data Uplink Activities The scenario starts with the Data Analysis and Interpretation Team (DAIT) retrieving the Clementine data image of the shadow edge area, where the rover is located at the start of the scenario. They review this image using their visualization system, represented in the Brahms model as a VisualizationSystem object. According to the work practice, they do this without anyone requesting that they look at the data. This means that the DAIT needs to know: 1) the location and situation of the rover at all times, 2) whether data is available and needs to be retrieved, and 3) where and how they can retrieve data.

Once the DAIT has retrieved the images, it communicates this to the Science Operations Team (SOT), and they collaboratively analyze these images (the AnalyzeRoverImages activity). When done, the SOT plans the first rover command sequence. According to the scenario being simulated, the SOT decides that the rover needs to drive for a specified amount of time (15 min) into the crater to a specific location (ShadowArea1InCraterSN1), and while driving it should be using its neutron detector instrument to detect hydrogen in the lunar surface. This decision is communicated to the Vehicle and Spacecraft Operations

Team (VSOT), as well as to the DAIT. After this communication, the SOT waits for the rover's downlink data.

Rover Activity. 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 agent. In the scenario model, the Neutron Spectrometer object is active and creates a HydrogenData_1 object containing the hydrogen data that is sent to Earth while the VictoriaRover is traversing to a permanently shadowed area within the crater SN1. The rover then waits for the next command sequence from Earth. During this time the teams on Earth are analyzing the hydrogen data and deciding what to do next. In the Uplink activity, the rover is given the command to search for water ice in the permanent dark area. This eventually triggers the drilling activity, which uses the SATM instrument.

To collect a sample the SATM has to 1) lower its augur to the surface, 2) drill to the depth given as part of the command by the SOT (in this scenario the command says to take a 1.0cc sample at 10cm depth), 3) open the sample cavity door, 4) continue to drill to collect the sample, 5) close the sample door when done, 6) retract the drill from the surface, and 7) deposit the collected sample on the instrument carousel.

In the Brahms model, the Augur object creates the LunarSample_1 object as part of its activity to capture the lunar sample, after opening the sample door and continuing the drilling to collect the 1.0cc sample. The activity times for drilling into the surface are dynamically derived during the simulation.

Downlink Activity. When the rover detects hydrogen in ShadowArea1InCraterSN1 the downlink process starts (represented by the Brahms AgentViewer in Figure 5).[3] The VictoriaRover agent contains the S-BandMGA object, which represents the S-Band transmitter on the rover. The VictoriaRover creates a data object with a) the current rover location information and b) the hydrogen data. This data object is then communicated to Earth, via the UsnDish1 object. The UsnDish1 object communicates this data to the DataConversionSystem, located at NASA Ames. As can be seen in Figure 5, 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 that the data conversion system interact with the visualization system without human intervention (details of the data conversion are not represented here).

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 (see Figure 5). When the DAIT agent detects that there is newly available neutron detector and location data, it retrieves the data from the VisualizationSystem object (the activities RetrieveNeutronData, InterpretNeutronData, and FindRoverLocationData).

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

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 sub-activity is to perform the DrillingActivity. Parameters indicate how deep to drill and how big a sample to collect at that depth. Figure 5 shows part of this second uplink process.

The duration of the downlink and second uplink processes determine the duration of the second DoNothing activity of the VictoriaRover, simulating the time the rover is waiting for the Victoria science team to decide the next command sequence.

---

[3] After the model is developed and compiled, the Brahms simulation engine executes the model in batch mode. A relational database is created, including every simulation event. An end-user display tool (AgentViewer) uses this database to display all groups, classes, agents, objects, and areas in a selectable tree view. The AgentViewer displays an activity time line of the selected agents and objects; communications may be optionally shown via dashed lines between agents and objects.

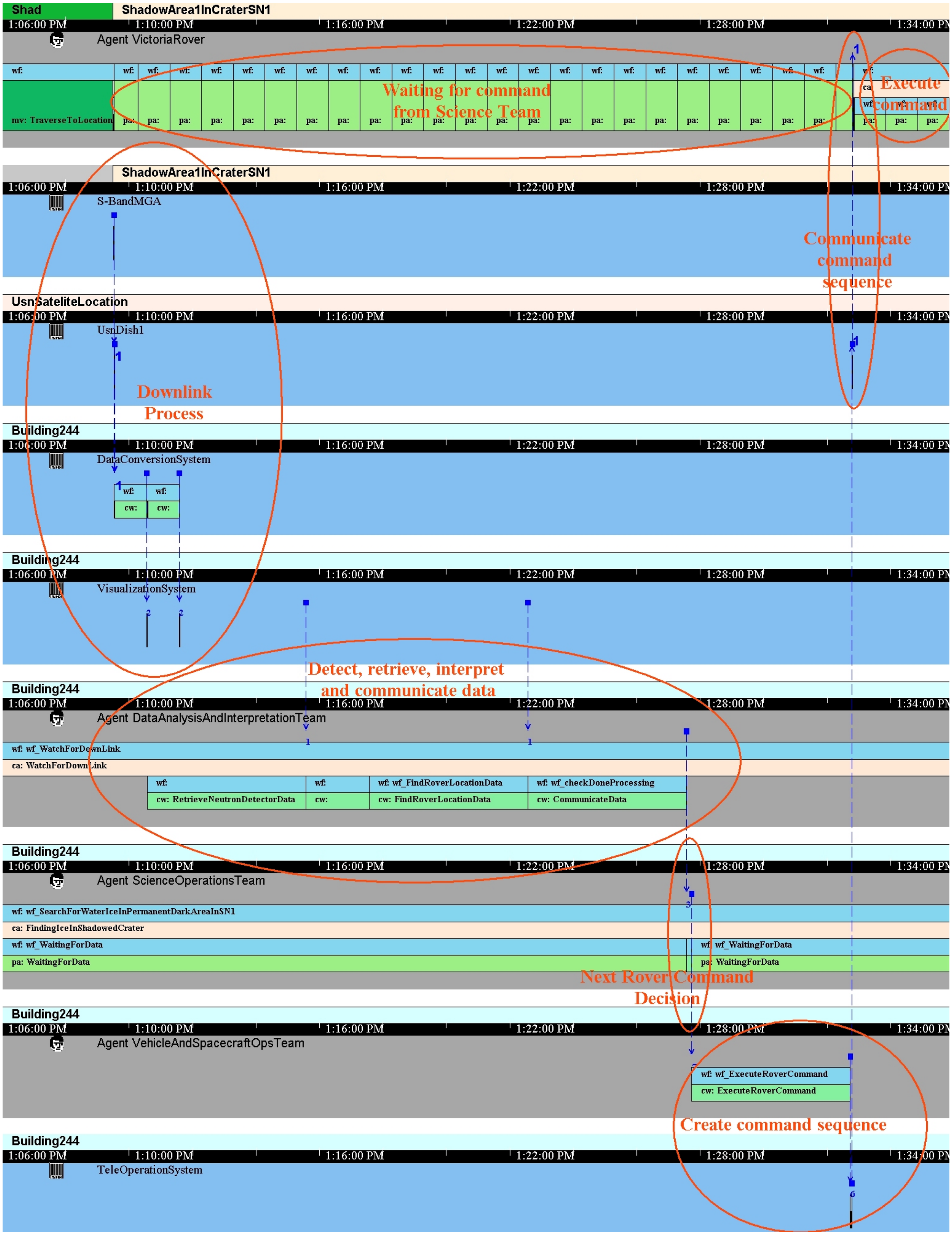

Figure 5. Simulation of downlink and second uplink command activities

USING CONCEPTUAL OBJECTS TO CALCULATE ENERGY USED

To calculate the total energy used by the rover, we need to represent in the model the energy needed for each subsystem during a rover activity. This is done using a conceptual object attached to appropriate workframes. The energy consumption for every rover activity during the simulation of the scenario is shown in Figure 6. In particular, the energy the rover uses during the *Waiting* activity (see "waiting for command from science team" in Figure 5) is defined by the energy needed for *Thermal Protection during driving + Command and Data Handling during driving*. While the rover is standing still and "doing nothing," it consumes power for its thermal protection and its commanding and data handling for its subsystems, such as its processor board.

Besides the power left to use after the scenario, another interesting variable is the energy usage rate by the rover.

$$\boxed{\mathit{EnergyRate} = \mathit{Total\ Power} \,/\, \mathit{Pbattery}(\mathit{start\ of\ traverse})}$$

Given the energy used in the scenario—drive 900m into the crater, and take one 1.0cc sample at 10cm depth—we calculate that the robot has used almost a third of its power:

$$\mathit{EnergyRate}(\mathit{drilling\ in\ permanent\ dark\ crater}) \approx 0.30$$

This variable represents the rover power consumption effectiveness of the simulated work system design, and is a measure that can be used to compare different work system designs for a model scenario.

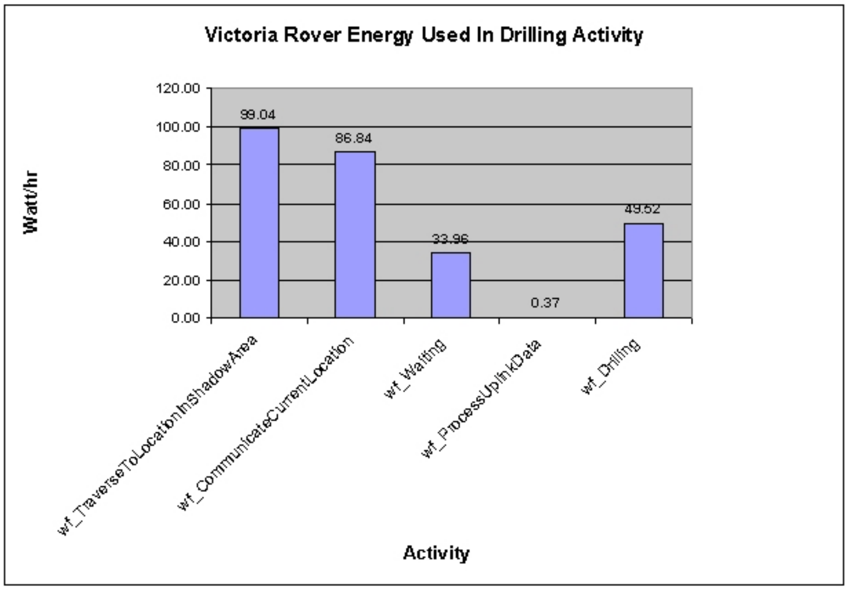

Figure 6. Rover energy used in high-level activities from simulation history database

Limitations of the Modeling Language

We believe that the Brahms language and simulation engine are just in their infancy, with decades of research required before we have accomplished our ultimate objective of modeling the complexities of human behavior in work settings. For example, we need to better represent the nature of identity as played out in interpersonal interactions (e.g., "office politics" and friendships); relate social, cognitive, and anthropometric models; model fatigue, boredom, diurnal rhythm, "external life" (e.g., errands, family interruptions); and model learning (especially by watching and mimicking). We also have practical challenges of developing reusable model components organized by types of settings and human interactions. To use Brahms for exploring a variety of workload conditions, it would be useful to have tools for statistically generating cases for simulation analysis. More broadly, we require theoretical frameworks for validating analog models (e.g., relating Arctic expeditions to Space Station experience and planned missions to Mars). In subsequent sections, we describe in more detail some of our immediate concerns for modeling NASA missions.

MULTIPLE-DAY SIMULATIONS

All simulations we have constructed to date have modeled behaviors over a few hours at most. In practice, we need to model at least a week of simulated time in order to show the rhythm of life and work. For example, it is common for experiments ("payloads") on the Space Shuttle to require more time than expected, carrying over into multiple days, and changing previous schedules. Understanding and modeling how plans are revised, represented, and communicated is a central part of work practice research. Modeling a Shuttle mission requires modeling 10 to 14 days; a Space Station Expedition lasts for several months; a mission to Mars will require about three years. Although various work-arounds are possible, we believe it will be necessary to extend Brahms to make it convenient and tractable to create long-duration simulations. The key problems are time-indexical beliefs, forgetting, and pattern detection, which we discuss here.

Many beliefs are time-indexical, that is, the meaning changes over time. For example, "the target selected for the rover last week" depends on the current time. Obviously, having a memory of past events is also necessary. Other beliefs refer to intentions, such as "the activity I plan to do this afternoon." In general, a model must be written from the start to allow time-dependent beliefs. For example, "the target selected for the rover" is part of a plan, and the belief must record both the time of this planned event and when the belief was generated.

If agents automatically have beliefs about the activities they perform, the requirements for memory would grow enormously (the effect on performance is less because Brahms uses an optimized reasoning state network). One approach is to declare certain activities as "reflective" (i.e., "cognitively penetrable"), which would restrict what beliefs about activity events (and referenced objects) are automatically recorded.

Forgetting should be simulated. People naturally forget; it is not necessary for the history of all events to be recalled even from week-to-week. Consolidation and abstraction of beliefs is necessary. However, most cognitive research on human memory focuses on how information accumulates, not how it is forgotten. Further, situated cognition theories suggest that remembering is a form of theorizing, not merely retrieving facts (Clancey, 1997). Trends and exceptions are remembered, but not routine happenings, which are blended and "anchored" by early experience. Crucially, forgetting depends on current activities. For example, an agent working on a particular task over several weeks may remember many details from the beginning (suggesting a possible hierarchical scoping effect). Although our interest in developing Brahms is fundamentally on simulating interactive behavior and not learning or reasoning per se, we must incorporate a model of memory if we are to simulate behavior over multiple days.

Repeated experiences should influence subsequent behavior. A simulated agent should not "mindlessly" repeat behaviors. People notice patterns and break out of loops. Also, people get bored or tired if forced to repeat behaviors. Pattern detection in experience (e.g., "This is the same process that produced an error yesterday") plays an essential role in learning, plus repetition implicitly influences motivation and level of attention. At another level, social theories of learning suggest that people learn by mimicking others (so co-located workers tend to learn about each other's jobs). Further, people develop relationships with each other, influencing their interest to assist each other, by being co-located. One simple approach to modeling learning of this sort is to have interactions between individuals in particular situations lead to an exchange of behaviors (workframes are exchanged). This is a straightforward application of existing work in cognitive science, with the proviso that we do not interpret this "transfer of expertise" literally, but view it as a modeling that people learn from each other. Furthermore, although much of cognitive science is concerned with modeling human learning, very little research has *modeled* learning behavior as interactive, interpersonal, and resulting from patterns detected gradually and incidentally.

MISSIONS, SCHEDULES, AND VEHICLES

In developing Brahms simulations, we have not previously emphasized the static class-instance descriptions one finds in conventional knowledge models (such as expert systems). However, such representational constructs are needed to describe mission and expedition scenarios as relationships between Brahms model components. For example, the work in a Victoria mission involves multiple shifts (a particular role is fulfilled by different people during the day), vehicle trajectories, and timelines of activities. More generally, a space mission scenario involves a description of groups, locations, objects, and activity plans (e.g., a Shuttle mission). Further, locations (of the Shuttle) and group membership (crew of the Shuttle) change during the course of a mission (e.g., exchanging crew members with the Station). Neither these static nor dynamic features have been adequately incorporated in the Brahms language. The notion of a "conceptual object" in Brahms (originally included to allow representing "job orders" in office workplaces) could be extended to dynamically represent a configuration of groups, agents, objects, locations, and time-stamped activities. Clearly, the notion of a *schedule* is basic and needs to be represented conveniently using an interactive, hierarchical editor (not as a list of beliefs). Some basic constructs are outlined here.

LOCATION-GROUP (LG): the people who occupy (live or work in) a certain location at a certain time. Notice how the groups in Victoria are idealized because they are defined by function, which is location independent. In contrast, consider the group, "people living and working in the Mars Arctic Research Station" (Clancey, 2001). This group changes during *phases* of an *expedition*, and may include a visitor on a particular day. Further, the location of an LG may change, such as "people living and working in the Space Station during Expedition 3"—the location of the Station changes every moment. Brahms currently provides no method for changing group membership (let alone the location of a building) during a simulation. In our original focus on office work, organizational changes were infrequent. In retrospect, we realize that office meetings and other projects are improvised during daily work and require the same capability to represent both planned and dynamically modified group membership.

SCHEDULED ACTIVITY-GROUP (SAG): a planned LG, e.g., a rotation or phase during an expedition, a particular Shuttle mission. More formally, a SAG is a *group* planned to engage in a particular *activity* at a particular *location* (or trajectory) for a certain *duration* or on certain start and end times. SAGs may be hierarchically nested, as a phase (with particular members) during an expedition. For example, "Clancey was a member of Rotation #2 inside the Arctic Research Station from July 8-17 during the Haughton-Mars Expedition for the 2001 field season." SAGs may be planned, active, or past. SAGs often occur in a series, such as shifts for work day, which may or may not overlap. Group roles repeat during every SAG in a series (e.g., each Station crew has a commander). Agents may be *temporary members* of a planned SAG, e.g., a visitor on the Station during an expedition. A SAG usually has planned (and often written) activities on a timeline (a *schedule*).

LOCATION-OBJECT (LO): objects in which people live, whose location changes over time, e.g., the Space Shuttle, a "Transhab" spacecraft for going to Mars from Earth, a pressurized Rover on Mars. Brahms development originally focused on office work in cubicles; in shifting to NASA's world, we must model vehicles, space bodies (planets, satellites), and trajectories. Objects in space have combined properties, some of which change over time. For example, the Space Shuttle is a *vehicle*, which becomes a *spacecraft-in-orbit*, which is a *satellite* that is a *habitat*. Our original notion of Brahms geography model as consisting of rooms in buildings in a city seems humorously simplistic. In effect, some Brahms objects must be also "area definitions," such that agents and objects can occupy them. This extends the object-oriented scheme to the geography model, so spaces such as rooms and buildings (and especially habitats and spacecraft) are modeled as three-dimensional objects with attributes and behaviors.

A NASA mission can then be defined as a SAG associated with one or more LOs. For example, mission STS-104 involves a Shuttle crew (a group), a particular Shuttle Vehicle (object), a Trajectory Plan (a kind conceptual object?), and Activity Plan (which might involve the Space Station). Victoria is a mission involving many teams, a rover, trajectories on the moon, and an activity plan for several months of lunar surface operations.

HUMAN BODY MODEL

In practice, where agents perform an activity partly depends on available space and tools. For example, an crew member in the Mars habitat may read in his/her stateroom if there are no comfortable chairs available. So modeling the activity of reading involves modeling chairs, a resource the agent requires. Similarly, the simulation display must be realistic, so the agent has a different visible posture when sitting in a chair. Further, the agent's zone of perception must relate to posture (e.g., standing on a ladder in the Mars habitat, one can look into the tank of water above the staterooms and determine the amount of water available). Here is a basic outline of considerations.

- Postures
  - Agents have postures, e.g., sitting, standing, lying down.
  - These postures occur on some surface or object, e.g., sitting on a chair, standing on a ladder, sitting on the floor.
  - Body posture is oriented with respect to other objects, e.g., facing someone else, facing the galley sink.
  - Postures may be composed: sitting at a table (by sitting on a chair that is next to the table).
- Zones of perception
  - Line of sight, e.g., facing the galley sink, an agent cannot see who is standing on the ladder; looking outside the West portal, the agent can see the airport runway
  - Within earshot, e.g., a whisper on the lower deck cannot be heard on the upper deck
- Moving with someone or something
  - An agent or object follows (or keeps constant distance from) another agent or object, e.g., the Robotic Assistant moves with the astronaut, the crew member follows the commander in the EVA preparation room.
- Carrying contained objects
  - Contained objects are brought along, even when the agent doesn't know what is inside, e.g., a robot carries a box and the contained objects change their location, too.
- Incremental movement
  - Movement is discrete, so the agent/object is located at different places along a path over time.
  - Interactions may occur between the agent and the objects in the environment during the movement (e.g., having a conversation with someone you encounter).
  - Movement may be hindered or varied in speed by other objects in the environment (e.g., someone else is on the ladder, so you must wait for them to go up or come down).

OTHER FACTORS NOT CONSIDERED IN BRAHMS

In the 1970s and 80s, cognitive scientists commonly said that "the model is the theory"—the simulation embodied all of the factors and principles they understood to be relevant to human cognition. Such claims were especially possible because psychologists and artificial intelligence researchers almost unanimously assumed that textual components (e.g., "frames" or production rules) in simulations mapped onto physical structures in the human brain (e.g., an expert system rule not only represented an expert's knowledge, it was how knowledge was stored in the expert's brain). However, in Brahms we emphasize that we are modeling behaviors and not knowledge per se, so there is no necessary relation between

Brahms constructs and how the brain works. As we incorporate aspects of memory and learning, we must of course make such commitments; but even then we will not suppose to model how memory works, only its behavioral effects.

The distinction we have drawn represents a significant shift in how models are interpreted. Most importantly, we can now list many theoretical notions that are not embodied in Brahms models. The model is a pale reflection of our understanding, but hopefully a useful tool for designing work systems and training. Beyond the representation of memory, learning, perception, and postures, we have not worried about other well-known factors in human behavior, such as hunger and fatigue. We have not incorporated anthropometric models of reach and line of sight (e.g., sitting in a chair can a person reach a control switch?). At another level, we have only begun to model social relations and their effects.

Crucially, a Brahms model is not based on traits, in which "properties" of people interact. Rather, we model and study how *behaviors* interact in a simulated environment. Trait-based models parameterize behaviors through isolated properties (e.g., Bill is friendly) and state rules for how they influence agent behavior (e.g., two friendly people have longer conversations). In Brahms, such attributes would be represented as *relations* (e.g., Bill is a friend of Maarten) which conditionally influence behaviors (e.g., If you need help and agent X is your friend, communicate with agent X about your needs). Emphasis is thus placed on who knows whom and what people know about each other, rather than isolated attributes (e.g., an agent's skills). Modeling relationships, their influence on work practice, and how relations and behaviors change over time is a major research area for Brahms-like simulations.

To summarize well-known aspects of human behavior that are not modeled in Brahms:

- *Actual language* used by agents when communicating (e.g., how social conversations become task oriented)
- *Learning* by watching others or being told how to do something.
- Agents' *models of their history and trends of their group*: history of the group, competitive pressures, management's initiatives, changes in customers.
- *Cumulative effects of work flow*, especially the effects of continued interruptions and waiting (also: forgetting, variety, rhythm, fatigue, anxiety, exuberance).
- *Reconceptualization* (learning on the job) influencing later priorities, attitudes, judgments in handling difficult situations
- *Complex juggling and simultaneity of activities* to ensure closure, to be productive (e.g., reading while on the phone).
- *Life away from work*: breaks, vacations, family.

Each model we construct is 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 remains to be seen.

References

Bagnara, S., Zuccermaglio, C., and Stucky, S. (eds.). 1995. *Organizational Learning and Technological Change (Papers from the NATO Workshop, Siena, Italy, September 22-26, 1992)*. Berlin: Springer-Verlag

Cabrol, N. A., Chong-Diaz, G., Stoker, C.R., Gulick, V.C., Landheim, R., Lee, P. et al. (in press). Nomad Rover Field Experiment, Atacama Desert, (Chile) I. Science Result Overview. *Journal of Geophysical Research*.

Clancey, W. J. 1997. *Situated Cognition: On Human Knowledge and Computer Representations*. New York: Cambridge University Press.

Clancey, W. J. 2001. Field science ethnography: Methods for systematic observation on an Arctic expedition. *Field Methods*, 13(3):223-243, August.

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.

Corbett, J. M., Rasmussen, L.B., and Rauner, F. 1991. *Crossing the Border: The Social and Engineering Design of Computer Integrated Manufacturing Systems*. Artificial Intelligence and Society Series. Great Britain: Springer-Verlag.

Greenbaum, J., and Kyng, M. (eds.) 1991. *Design at Work: Cooperative Design of Computer Systems*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Sierhuis, M. 2001. *Modeling and Simulating Work Practice; Brahms: A multiagent modeling and simulation language for work system analysis and design*. Ph.D. thesis, Social Science and Informatics (SWI), University of Amsterdam, SIKS Dissertation Series No. 2001-10, Amsterdam, The Netherlands, ISBN 90-6464-849-2.

Wall, S. D., and Ledbetter, K. W. 1991. *Design of Mission Operations Systems for Scientific Remote Sensing*. London: Taylor & Francis.

Wenger, E. 1998. *Communities of practice: Learning, meaning, and identity*. New York: Cambridge University Press.