Human/Environment Relations Analysis & Simulation Using Human-Centered Systems Methods for Design and Evaluation of Complex Habitable Environments

by

Simonetta Andrea Rodrigùez

B.S. Architectural Engineering
The University of Texas at Austin, 1988

Submitted to the Department of Civil and Environmental Engineering in Partial Fulfillment of the Requirements for the Degree of

MASTER OF SCIENCE IN CIVIL AND ENVIRONMENTAL ENGINEERING
at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

January 2000
[February 2000]

Signature of Author: ______________________________
Department of Civil and Environmental Engineering
January 24, 2000

Certified by: ______________________________
Steven Lerman
Class of 1922 Professor of Civil and Environmental Engineering
Thesis Supervisor

Accepted by: ______________________________
Daniele Veneziano
Chairman, Departmental Committee on Graduate Studies
Civil and Environmental Engineering

Page intentionally left blank.

Human/Environment Relations Analysis & Simulation Using Human-Centered Systems Methods for Design and Evaluation of Complex Habitable Environments

by

Simonetta Andrea Rodriguez

Submitted to the Department of Civil and Environmental Engineering, on January 14, 2000 in partial fulfillment of the requirements for the degree of Master of Science in Civil and Environmental Engineering.

ABSTRACT

An exploration with two main goals is presented:

1. Modeling a self-contained "intelligent building," the Mars Arctic Research Station (MARS) habitation unit, and interaction with its users. The modeling effort seeks representation of invisible aspects of the interactions between the station and its users rather than visible forms.
2. Examination of how (or if) the modeling process contributes to understanding of the interactions between human beings and the structures and technological features of their habitations... particularly when something goes wrong.

The modeling language and associated software chosen for study in this exploration is the Business Redesign Agent-based Holistic Modeling System (Brahms), developed by a team of researchers and computer scientists now involved with the joint NASA/Mars Society MARS project.

The issues and factors relevant to any attempt at modeling human behavior in interaction with habitable environments are specified. Brahms is described as a potential human-environment-modeling framework and language, and separately as a software system. The process of developing a Brahms model of the MARS habitat and a comprehensive analysis of that process are presented. Recommendations for similar future work with Brahms, as a modeling framework and language, and separately as a software system, are provided.

Thesis Supervisor: Steven Lerman
Title: Class of 1922 Professor of Civil and Environmental Engineering

Acknowledgements

First and of greatest import to the author, is acknowledgment of the penetrating influence of the tenets of the Bahá'í Faith, learned in childhood, particularly the uniquely Bahá'í principle that *work*, performed in the spirit of ***service to humanity***, is actually *worship*, and that intellectualism or scholarship engaged solely for personal diversion or amusement is unworthy of the honor, power and grace invested in every human spirit.

Of scarcely lesser importance, is the profound example set by my mother, Seraphine Camino Rodriguez McLaughlin, who studied computer science long before any university in the world offered this field as a regular course of study, who went on to do what she was explicitly told she could not, that is, enter a profession "reserved" for a gender and color other than her own. Her loving support, guidance, gentle detachment and encouragement have been utterly priceless throughout my life, but even more critical in these years at MIT.

To my daughters Kiele Lokahi, "Fragrant Blossom of Unity," and Kelana Seraphine, "Two Dreams of Beauty," who have been most directly impacted by and involved with the vicissitudes of my life as a graduate student, I wish to say: I love you, and thus recommend that you finish school before your children are born, if possible... but remember that if this is not possible, the children themselves are inspiration and motivation to learn... as you have been.

To my husband John Saxton MacCord: I love you; "thank you" is an insufficient pale ghost of my feelings, and now, I get to learn with you what life is like "after MIT."

Many other people and organizations made this impending graduation possible.

By being "watering holes," that is, oases in the Serengeti plains that MIT can feel like to one who comes from a completely different "place": Margo Daniels Tyler, Dan Langdale, Steven Lerman, Ayida Mthembu, Mary Ni, Nadine Champagne, Mary Rowe, Cynthia Stewart, Isaac Colbert, Toni Robinson, Lynn Roberson, Sandy Pentland, Dr. Bethany Block, Ken Stone, Fred Cote, Jane Gould, Joan McCusker, Jessie Williamson, Kenji Iwamoto, Blanche Staton, Patricia Gercik, as well as others.

By "being there" even if not here: Manuelita Camino McCleod, Linda Christine Rodriguez Price, Dr. Roger Bengtson, Dr. Dan Wheat, Charles Garner, Dr. William Sims, Dr. Alan Hedge, Dr. Marion Finley, Dr. Walter Kroner, Dr. Mikio Shoji, Dr. Christopher Loretz, Dr. Hiroshi Furuya, Ms. Yoko Kawasaki, Dr. William Clancey, Maarten Sierhuis, Ron van Hoof, Rochelle Roberts, as well as others.

By providing funds: IBM, the College of Human Ecology at Cornell University, the State University of New York, the IFMA & Haworth Foundations, the National Science Foundation, the MIT Graduate Education Office, the MIT Department of Civil & Environmental Engineering, the National Aeronautics and Space Administration, John S. MacCord, Seraphine McLaughlin, and my daughters by doing without.

By being friends... you know who you are, I thank you.

Table of Contents

1. Introduction and Context

Scheduled for early summer this year (2000), an unusual construction project shall begin within a large impact crater, Haughton Crater, on Devon Island in the Canadian Arctic (Lee 1999). A relatively small habitable structure will be built—or deployed—in the crater, to house would-be explorers of another world. The planned structure and the activities of its inhabitants highlight a collaboration between the Mars Society[1], a non-profit organization which advocates exploration of the planet Mars, and the U.S. National Aeronautics and Space Administration (NASA)[2]. The joint project is called the Mars Arctic Research Station (MARS)[3]. Documented in this thesis is another kind of exploration, with two immediate goals:

Figure 1.1 — Haughton Crater[3]

1. Modeling a self-contained "intelligent building," the MARS Station, and its interaction with its users. The modeling effort sought to represent invisible aspects of the interactions between the station and its users, rather than depictions of the station's physical form and appearance.
2. Examination of how (or if) the MARS station modeling process contributes to understanding of the interactions between human beings and the structures and technological features of their habitations... particularly when something goes wrong.

1.1 General Context

Certain buildings on the surface of the earth share important characteristics with specialized life-supporting transport systems such as submarines, airplanes, ocean or space ships, and space stations. Hospitals, jails, schools for the disabled, nursing homes, mid-ocean oil drilling

---

[1] http://www.marssociety.org/

[2] http://www.nasa.gov/ and http://www.arctic-mars.org/

[3] http://home.marssociety.org/arctic/ © 2000 The Mars Society & NASA. Used with permission.

platforms and polar science or exploration stations share these characteristics of interest with such transport systems:

- These are habitable environments serving as comprehensive life-support machines for short or long terms; the inhabitants are dependent upon the structure and its associated equipment and personnel for basic life-support, to some extent or totally.
- The inhabitants are not free to leave on momentary notice.

The association is odd, because the operators/inhabitants of submarines and space ships are able-bodied and trained personnel, while the inhabitants of hospitals, jails, and schools are unusually vulnerable due to ill health, childhood, or imposed restriction. However, in each of these environments, whether the vulnerability is due to the attributes of the external environment, or due to the attributes of the inhabitants, human lives depend upon the structure—the life-support machine—and its associated equipment and personnel, to perform life-support functions at all times, without failure.

1.1.1 Problem

Envisioning a future in which all of today's building stock has been replaced by newer buildings, it is certain that each of those newer buildings will include computer technology is part of the basic infrastructure performing life-support functions. At present microprocessor-controlled systems are becoming ubiquitous in all parts of the typical building: the heating ventilation and air conditioning (HVAC) systems, fire and other safety systems, elevators, locks and other access control systems, all are rapidly being transformed into computer operated devices. The trend will accelerate, as the development of computer technology accelerates. While a large building is in fact already a complex machine, in the envisioned future, the building machine will also be capable of much greater "self-directed" or "adaptive" activity than is typical for buildings today. The term "intelligent building" has been applied since the mid 1980's to buildings that really are rather not. (Caffrey 1988, Beck 1993) Yet since computing power is promised to show up in our clothing and rooms, making them "smart clothes" and "smart rooms" (Pentland 1996, 1998),[4] clearly, computer systems will drive the buildings of the

[4] http://sandy.www.media.mit.edu/people/sandy/

future to "do" more than today's buildings do. It is also clear that they will be critically involved in all of the systems that support human life. Similarly, transport vehicles such as space stations, do now and will continue to incorporate computer technology, including in systems directly supporting the lives of the human inhabitants.

1.1.2 Why is this interesting?

The question is not whether failures will occur in the computer system, or any other particular sub-system of the life-support machine, because failures will occur. The interesting question is, when a failure occurs, how will the human beings and these "intelligent" systems interact? For that matter, during normal operations of complex, "intelligent," life-supporting habitable structures, how will human beings and these systems interact?

In the motion picture *2001: A Space Odyssey*, a computer named Hal is depicted as possessing the power to deliberately end the lives of human beings, albeit while the machine was operating in a severely malfunctioning state after human beings had lied to it. One year before the date of the scenario portrayed in the film, computers are much more ubiquitous than the filmmakers anticipated. Rather than one large computer in any given setting, today there are dozens, hundreds or thousands, many behind the walls and imbedded within hidden systems in buildings. None of the computers in use today outside of research environments can, as did Hal:

- recognize and exhibit emotions, Picard (1999)
- deliberate,
- exercise self-survival maneuvers for which they are not programmed,
- handle arbitrary events for which they are not programmed,
- process natural human speech in real time,
- speak naturally,
- see as do humans or animals,
- hear as do humans or animals,
- react coherently to anything for which they were not programmed.

"Intelligent rooms" such as Hal provided in the film have been the subject of intensive research (Coen 1998, 1999; Pentland 1998), yet the state-of-the-art in this research area does not provide the flexibility, versatility or comprehensive human-like intelligence of Hal.

However, today's hordes of small, limited, inflexible computer systems are rapidly moving into life-support roles similar to Hal's. While it is unreasonable to expect behavior like Hal's from real computers today, it is reasonable—perhaps essential—to expect that engineers should *engineer preventative measures* for disastrous human/computer/environment interaction scenarios similar to that portrayed in the film.

1.2 Human Context

Illuminating the problem requires some information about human beings, data acquired in decades of research effort in the field of environmental psychology, also called human/environment relations, and the more general fields of psychology and education. Important and relevant constructs from such literature regarding human beings include situation awareness, learning, error, cognitive processing, working memory limitations, work practice, decision-making, situated cognition, and more. These are not absent from this discussion because they are unimportant, but because detailing them all is not the focus here. Two important human factor constructs are sufficient to illustrate why the human-machine or human-environment interaction in failure situations is interesting and important to understand: *stress*, and *control*.

1.2.1 Human Factors

1.2.1.1 Stress

While everyone experiences stress in everyday life, the concept is typically discussed imprecisely. To understand the impact of stress on human interactions with complex systems, some level of precision for the term is required. Webster defines stress as "a physical, chemical, or emotional factor (as trauma, histamine, or fear) to which an individual fails to make a satisfactory adaptation, and which causes physiologic tensions that may be a contributory cause of disease". Cohen *et al* (1986), use the term stress "to refer to the study of situations in which *the demands on individuals tax or exceed their adaptive capabilities*." (emphasis added) The contribution of prolonged stress to disease is well-documented; the most relevant aspects here are the clear indications that 1) stress causes physiologic, often deleterious, changes in individuals, and 2) stress, as a condition, occurs when an individual fails to adapt to impinging stress factors.

The adaptive response to stressors is the subject of significant research. Evidence indicates that the adaptive process, also called the coping process, entails costs, which are "deleterious effects of an encounter with a stressor." (Cohen *et al*, 1986) While a stressor may cause direct negative effects on an individual, the evidence indicates that *additional* damaging effects occur indirectly from the process of coping with the stressor.

The profound effect of stress on the performance of human beings in various settings, especially complex and demanding settings, is well established. While stress is most often viewed as a negative condition, a minimum level of physiologic tension is essential for optimal performance in certain settings, since lack of tension can result in loss of vigilance and attention. Hutchins (1996) documents aspects of stress as they effect military operations involving management of "complex, multi-task situations" and technologies.

1.2.1.2 Control

Human or personal control is a variable of perception, the sense of persons that their actions potentially can influence their environment. (Cohen *et al* 1986) A large literature exists on the importance of control in human behavior, as control seems to be a deep-seated aspect of human functioning. (*Ibid.*) The human capacity for coping with environmental stressors is affected by personal control. While the literature on this subject is complex, because the details of the effects are complex, it is sufficient here to note that in general, people are better able to cope with stressors if they perceive that they have greater control. It is well established that both stress and lack of control affect the physical well-being of human beings, in addition to affecting such managerial work-place concerns as productivity, performance and job satisfaction.

To illustrate the role of human control in buildings, note that windows in older buildings, before the introduction of modem heating, ventilation and air-conditioning (HVAC) systems were always operable. Few modern office, commercial or industrial buildings since the 1950's permitted occupants to open the windows. The engineering rationale for non-operable windows is obvious: one can not design an efficient HVAC system if the engineering characteristics of the inflows and outflows to the system are not known at all times. Operable windows necessarily introduce large variability in crucial indoor-air inflow and outflow characteristics such as temperature, humidity and ventilation volume.

Yet the very concept of a "sick" building is a modem concept, the problem arising with the advent of modern HVAC systems and sealed windows, especially since building ventilation standards were drastically reduced after the 1973 energy crisis. (Hedge, 1992, b) Studies have shown that a building can be considered "sick," with many or even the majority of its occupants experiencing problems they consider related to the building, but with no identifiable physical causal factors. After over a decade of research seeking to identify the environmental causes of "sick building syndrome" in many buildings labeled in this way, it is now clear that psychological variables are essential to understanding the phenomenon. (Hedge a at, 1990. 1993, 1995) Thus with respect to windows in buildings, it appears that in modern buildings efficiency was gained through engineering control over the indoor air environment by sealing the windows. However, human control was lost, which operates to diminish human comfort and health, even when all mechanical systems are functioning as best as can be designed by engineers.

1.2.1.3 Failure As A Human Stressor

It seems obvious that: 1) When something goes wrong, people get upset, 2) When something *major* goes wrong, people panic. Adults acquire this common-sense knowledge through decades of life-experience. A failure condition in a complex, life-critical, technological system, is obviously a stressor for individuals who know of the condition or are responsible in some way for operation of the system. Evidence from the human environment literature indicates that the degrees of stress perceived by the human beings involved in a complex system are likely to be proportional to their perception of the importance of the failing system, that is, proportional to the perceived threat to life safety. (Cohen *et al* 1986) A variety of reaction behaviors to a failure condition and its accompanying stress is expected, depending on such factors as knowledge, training, "common sense" and individual psychological characteristics. This is of course why astronauts, submariners and pilots are superbly trained. Aside from enabling them to accomplish the tasks expected of them in normal operational modes, intensive training is expected to reduce the variability of reaction behaviors to failure modes.

Computers, however, have no equivalent to common sense, and all the knowledge imbedded in them must, at present, be programmed into them in one way or another. "Training" of neural nets, for example, is really a type of programming. To get back to the interesting question posed initially, we can re-phrase it as follows: When failure occurs, how will the human beings—who have changed their physiological state, since the failure acts as a stressor—and the computer systems—which can not change their state unless programmed in some way to do so—interact? Is there an inherent threat to life safety lurking in the idea that failure conditions can be exacerbated and driven further toward failure by computer systems that have no "awareness" of the emotional and psycho-physiological stressed status of the humans with which they interact?

Addressing these and similar questions demands greater attention to parameters of human beings –e.g. socio-biological research – than has been common for information technology systems in buildings. In part, this is because installation of information technology systems in buildings has so far been almost entirely limited to "showcase" office-building projects. All extreme failure conditions in such buildings are handled by first evacuating the structure, and *then* dealing with the building as a broken machine, devoid of people.

When most of today's stock of habitable space is replaced by structures chock-full of complex imbedded information systems, some structures will not be so readily evacuated: hospitals, nursing homes, jails, pre-schools, mid-ocean drilling platforms, all spaces occupied by vulnerable populations. Addressing this problem—the interaction of vulnerable human beings and complex imbedded information technology systems providing life-support functions in habitable spaces—is the goal of the research discussed in this thesis.

1.3 Infrastructure Context: "Intelligent" Buildings

So-called "intelligent" buildings are not new. The first building project explicitly labeled "intelligent," CityPlace Center in Hartford Connecticut, was completed in 1983 (Gannes 1984). This office building was designated "intelligent" because it included a building automation system (BAS), with timers and sensors to turn out the lights and monitor the temperature in offices, and a shared tenant telecommunication system, a phone system programmed to chose the least expensive routes for long-distance calls. Unfortunately, this use of "intelligent" as an adjective to describe a building created a worldwide misconception that this building had an electronic "nervous system", with walls that could "see" and "hear", and a "brain" that could "think", and this misunderstanding of the term intelligent building persists today. CityPlace was constructed as a showcase for the suite of building supply and automation products sold by the parent construction company.

1.3.1 Successful Marketing Tool

As a selling tool, CityPlace Center was remarkably successful, as this timeline of the period 1983-1992 shows:

- 1983: First "intelligent building" (IB) opens: CityPlace, Hartford, Connecticut (Anonymous, 1992)
- 1984: Over 12 IB projects in the U.S.. 50 agreements to install intelligence packages in new buildings (Gannes, 1984)
- 1985: Five IB's under construction in Japan (Anonymous, 1992)
- 1986: 71 completed IB's in Japan (Anonymous, 1992)
- 1990: NEC Corp headquarters, Super Tower, "world's largest intelligent edifice" (Anonymous, 1992)
- 1992: 2,000 IB's constructed in Japan. (Anonymous, 1992) Many others in U.S., London, Shanghai, Singapore, Spain, Australia, Hong Kong, etc.

These projects represent the first generation of "advanced" habitable structures built on the planet. There is evidence that the first generation of such projects have been disappointing to those who built them and use them, for voices of discontent appeared early:

- 1989: "The vision.. runs far ahead of what is affordable and technologically feasible." (Herbst)
- 1990: "The intelligent building fever has abated for the time being." (Horitake)
- 1990: "What ever happened to intelligent buildings?" (Carlini)
- 1991: "Intelligent buildings: myth, reality or wishful thinking (Finley & Akimura).

Several questions naturally develop at this point. What common characteristics are found in the first generation of "intelligent" buildings? What is their logical character or theoretical basis? How have these projects been evaluated? How have they performed?

Previous technical design concerns for IB's have been so varied as to invoke the notion from human psychology of "multiple intelligences", since there are few common characteristics among the projects designated as intelligent. Small subsets of these projects possess commonalties, typically sharing no more than one of the following technological approaches to intelligence:

- Structural systems: resist earthquake movements by active mechanical or hydraulic effort. (Uras & Aktan, 1993. Senders, 1992) or monitor structural systems and communicate information about cracks, etc. (Robison, 1992; Huston & Fuhr, 1993).
- Communications and Information Management: integrated systems (Kelly, 1986).
- Materials: self-repairing concrete and "glass nerves" (Lee & Selkowitz, 1993).
- Lighting and HVAC: "smart envelope" consisting of glazing, tints, shading devices integrated and controlled by computers (Lee & Selkowitz, 1993); chillers managed by expert systems (Kreider & Wubbena, 1991).
- Noise abatement: sensors monitor external noise and actively control noise vibration (Johnson, 1994).
- Facility management: incorporate artificial intelligence into work processes (Weingarten, 1991; Fink. 1991;Takahashi & Kateeshock, 1992).

The logical or theoretical concerns for the design of IB's have also been extremely varied, with no consensus developed about what constitutes or defines "intelligence", nor the key elements that should guide the design of these projects. Many researchers and authors have attempted to provide guidance, with the idea of "control" most often represented in some form. While not stated so explicitly, control is understood in these contexts to be machine control. This may be due to the number of authors and researchers in the IB field who work for building control system firms, yet the industry has not adhered to any common set of these ideas as definitive. Some of these attempts are as follows:

- Japan's Intelligent Building Study Committee: an intelligent building comprises three logical groupings called "spaces", the intelligent space, the route space, and the accommodation space. (Tateishi, 1989).
- "Integration as the key to building intelligence." (Geissler, 1989, Mool, 1989)
- "Strategic information systems." (Kujuro & Yasada, 1993)
- Intelligent buildings as the evolution of facility management, (Kujuro & Yasada, 1993) focusing on the unified management of "people" and "things." (Umeno, 1993)
- The intelligent building must "think by itself." (Tateishi, 1989)
- "The heart of the intelligent building is the sensing and control system, the BAS [Building Automation System]." (Myers, 1997)

In summary, all previous forms of "intelligence" in buildings, in practice and in theory, (where there is any theory), appear to be manifestations of classical engineering control systems methods, incorporating no findings from socio-biological research. Previous evaluation concerns have been financial performance and marketability (Cross, 1985). It is interesting and important that IB's have performed poorly on both counts.

1.3.2 Lessons Learned From the First Generation

Obviously intelligent buildings are constructed to be occupied by people, and these occupants form the reason for the existence of the buildings. Yet the viewpoints of occupants are not represented in the literature on intelligent buildings during the fifteen year period, 1983-1998. What do the people occupying these buildings think of them? We do not know. The designers of the IB's do not refer to occupants as essential criteria in the design process, nor as essential criteria in the evaluation process.

Yet procedures to evaluate buildings from their occupants' perspectives are well known. One tool for this purpose is called Post Occupancy Evaluation (POE) (Preiser et al, 1988, 1991). A POE is more than a "customer satisfaction" survey. It is analogous to the series of structured questions a physician asks a patient during a medical exam. While phrased in layman's language, these questions provide specific technical information to the physician. A specific and exhaustive POE, calculated to provide data on whether and how the engineering design intentions for a habitable structure arc represented in the experience of its occupants, is possible and feasible. It seems obvious but is worth saying that data from such systematic investigation of the relationship between engineering design intentions and occupant experience would usefully inform the design of the next generation of IB's. (Wener, 1990) In later chapters, the relevance of POE to the project at hand, the MARS habitation unit, will be discussed.

1.3.3 What Is Needed

In the author's opinion, the research need is that these environments, whether already constructed or planned, must be examined anew with a *human-centered* mindset. Serious, meaningful data about occupants must be gathered, methods to collect such data must be developed, as well as methods to use the data for understanding occupants' interactions with structures and systems. Theories and models of human-computer-building interaction must be developed and tested.

In short, the community must *engineer* prevention of the disastrous "2001" human-computer-life-support interaction scenario. "Systematic evaluation works to the benefit of all who use buildings or are otherwise involved in their creation and operation." (Baird *et al* 1996) Computer simulation, as part of a comprehensive system of systematic evaluation, provide advantages over traditional experimental methods. (Gulyas *et al*, 1999). The possibility of human/building-system interaction simulation has not been previously addressed in the IB literature. Computer simulations of the interaction of human beings and complex intelligent life-support environments can and should be performed. This thesis represents a first attempt.

1.4 Computational Context

In addition to substantial research on simulation methods in many fields, relevant prior computational research include human-computer interaction (HCI), human-centered systems (HCS), and human-centered computing (HCC). While HCI as a discipline has developed a huge literature in the past 10-20 years, HCS and HCC are relatively new terms, essentially synonymous, defining a new approach.

1.4.1 Human-Computer Interaction (HCI)

A broad overview of HCI research must suffice here, for the field produces more papers *each year* than any one person can read (Neilsen 1995). HCI focuses on "behaviors exhibited by humans and computer-based artifacts while they are affecting each other," on "developing computer-based artifacts," and "testing a separable and controllable aspect of human-artifact interaction." (Blumenthal 1995). Some HCI researchers argue for purely empirical approaches to problem-solving in the field, essentially abandoning the AI-based and cognitive theories that have been the theoretical mainstays of HCI for some time, while others point out that such approaches can prove pointedly unsuccessful. (Nielsen and Bergman 1995) Bardram and Bertelsen (1995) note that, "Making better interfaces seems to be a goal shared by most people in the HCI community, but it seems hard to agree on what constitutes a good interface; and to what extent a scientific foundation for design is necessary."

The HCI field has in recent years begun to explore *activity theory,* a Russian approach to the issues of human-artifact interaction in social contexts, while not fully embracing the complete conceptual basis of the activity theory approach. (Blumenthal 1995). Space does not permit a thorough explanation of activity theory here, a tutorial is provided by Kaptelinin *et* al (1995). The large amount of research in the HCI field over the past 20 years appears to have provided the "shoulders to stand on" for a newer conceptual system, discussed in the next section.

1.4.2 Human-Centered Systems (HCS) & Human-Centered Computing (HCC)

The "human-centered systems" approach was first defined in a workshop sponsored by the National Science Foundation in February 1997 by researchers in the computing, social, behavioral, organizational, information, and engineering fields. The stated goal of the workshop was "to define this emerging multidisciplinary held and articulate research, educational, and infrastructure needs to support work in this area." Part of the published summary is quoted here:

> *Motivation: Why Support Human-Centered Systems Research?*
> *The concept of "human-centered systems"... represent a significant shift in thinking about information technology... that embraces human activity, technological advances, and the interplay between human activity and technological systems as inextricably linked and equally important aspects of analysis, design, and evaluation... Research in human-centered systems advances basic scientific knowledge in such areas as distributed cognition, speech, and social systems, in disciplines ranging from linguistics to psychology and computer science. In an era of unprecedented technological change and growth, basic scientific research is crucial to design appropriate interventions into complex human social systems and to analyze and evaluate the affects of such interventions.*
> *Definition: A system is defined as an agglomeration of interacting interdependent components which used in combination accomplish an activity that no one component can perform alone. In this report we focus primarily on information, communication, and distributed knowledge systems. A human-centered system aims to serve human activity. It is one that incorporates explicitly human (perceptual, motor, cognitive, and social) ramifications as components of design...*(continued)
> *Human-centered systems employ computing technology as a tool for the human user, not as a substitute; the human is the ultimate authority for control and the technology is employed to expand human capabilities and intellect...*
> *The human-machine interface enables users to acquire information, explore alternatives, execute plans, and monitor results. Making a high bandwidth interface that present data rapidly in a form that facilitates human decision*

making is the central challenge. Development of multimodal interfaces is... a central concern of human-centered systems...

It is insufficient to study and model people in isolation from technology or technology disconnected from a field of human activity. Both perspectives are needed in a fundamentally integrated way. An implication of this view is the centrality of field work to provide real data on real activity in real contexts. A related issue is metrics: how can we measure what is happening in a distributed cognitive system in a meaningful way. (Flanagan *et al* 1997)

It can be readily questioned whether the "significant shift" argued at the beginning of this passage is real. Have not many researchers and businesses concerned with information technology systems focused on human beings for many years? Hardware and software systems explicitly designed with human beings in mind are not new, a short list of such items includes graphical user interfaces, mice, trackballs, touch pads, touch screens, computer-based learning systems, computer-aided design, distance collaboration, and so on. Furthermore, the vast "Total Quality" movement, several decades old, includes such notions as "quality is customer satisfaction", "voice of the customer", and so on, all of which are clearly focused on human beings, the "end user." Are Flanagan *et al* re-hashing long-established definitions, conjuring something that appears "new" in order to justify funding arguments and the like?

While focusing on the needs of the fabled "end user" is not new, some fields have done a better job of it than others. The contention here is that in the case of information technology in habitable spaces, i.e. "intelligent" buildings, there has been little or no systematic focus on human beings as users and occupants of these structures. Thus in *this* field, NSF's "new" approach is indeed new.

1.4.3 NASA and HCC

The National Aeronautics and Space Administration apparently prefers the term "human-centered computing", and defines HCC as a "research methodology for analyzing, designing and/or evaluating AI systems and other computer-related technologies within a human biological, cognitive and social context-" (NASA 1998c) An illuminating comment about the HCC approach appears on the NASA Ames Research Center web site:

> *Computer scientists generally share the public's excitement in having the latest and greatest technology, whether it be a large laptop screen or a wireless communicator (such as a cellular phone). The difference of course, is that computer scientists not only use the tools they build, they are involved in inventing new kinds of hardware and software. This process of invention may, broadly speaking, proceed from two directions; One may start with theoretical possibilities and see what you can build (such as a faster computer or a software program that automates routine work). Or you may start with people, study how they do their work, and understand what tools would help them. Human-centered computing starts with this second approach—we observe people firsthand, talk to them, and invite them to collaborate with us in inventing new computer tools.* (Clancey, 1998)

It was explicitly stated on NASA Ames' web site (in 1998) that HCC is a "Major Research Focus Area" for NASA (1998c). HCC as it applies to the specific context of the MARS endeavor is discussed on the project's web site[5]. A tool arising from NASA's HCC focus is the modeling and simulation system named "Brahms," discussed extensively in the remainder of this thesis.

1.4.4 Thesis Outline

In Chapter 2, issues that should be addressed in modeling or simulation efforts of human-environment interaction are delineated.

In Chapter 3, Brahms is described as a modeling language and as a software system.

In Chapter 4, modeling of the MARS Station using Brahms is described.

In Chapter 5, the modeling and simulation process and its resulting output is analyzed. The chapter concludes the thesis with recommendations for future work.

[5] http://www.arctic-mars.org/SCIENCE/hcc.html

2. Analyzing Human-Environment Interactions

2.1 Analysis and Modeling Issues

As stated in Chapter 1, the research presented in this thesis involves analysis of the interactions between human beings, built environments and technological features, especially computer technologies. The research incurs the notable difficulty that three distinct professional and intellectual disciplines intersect in such analysis. These are: a) the social sciences which focus on human behaviors and social interactions, b) the architectural and engineering sciences with focus on the built environment, and c) the computer and information sciences which focus on control technology and human-computer interfaces. The issues of interest to each of these disciplines vary enormously when analysis is needed for a particular project.

Figure 2.1 (next page) lists issues that just two scientists of these intersecting disciplines would consider relevant for most conceivable habitable-space projects. The two scientists are, a) an environmental psychologist, perhaps a member of the Environmental Design Research Association [1] and, b) a facilities planning and management researcher, perhaps engaged in research sponsored by the International Facilities Management Association.[2] Of the issues listed in Figure 2.1, some were discussed in Chapter 1, some will be referred to in the remainder of this thesis, but space does not permit detailed discussion of each, or why it is included in the figure. They are all relevant to the problem space at hand, however, for experienced practitioners of these two professional disciplines.

To help illuminate the issue, consider the MARS project introduced in Chapter 1. Since the project is entirely scientific, data collection of all types is its mission. Assume that an interdisciplinary subset of the project's scientists specifically focuses on the human-computer-environment interaction issues discussed here. They decide beforehand that data should be collected throughout the habitation period and install appropriate sensors and recording devices to collect this data. With current research capabilities and technologies, it is possible to collect significant quantities of data for each of the issues in Figure 2.1.

[1] (known as edra) http://www.telepath.com/edra/home.html
[2] (known as IFMA) http://www.ifma.com/

Fundamental Physical Constructs:

Agents: Persons, AI Systems.
Habitable Spaces: Buildings, Rooms, Campuses (Design, Construction)
Paths: Corridors, Doors, Streets, Stairs, Walkways, Elevators, Hatches
Controls: Environmental, Personal
Objects: Devices, Appliances

Important Non-Physical Constructs:

Groups & Hierarchies: Human, Objects, Classes
Safety, Errors: Routine, Catastrophic (21st Century Life Safety Codes?)
Costs: Capital, Life Cycle, Resale, Energy Usage
Time, Scheduling
Satisfaction, Comfort
Deployment, Construction, Adaptations, Repairs
Maintenance, Reuse, Decommissioning
Ownership: Financial, Space Domains

Some Issues Relevant to the Constructs Above:

Agent:

Person: Emotion (Affect, Pleasure, Fear, Anger, Anxiety, Confusion, Depression)
Stress (Psychological, Physical), Health (Psychological, Physical)
Capabilities (Physical, Learning, Adaptive, Languages)
Beliefs (Self, State, Group Identity, Capability, Control)
Actions (Work, Play, Self Care, Sleep, Movement, Communicate)
Engagement, State

AI System: States, Capabilities, Actions

Spaces:

Size: Height, Width, Length, Area, Volume

Shape: Orthogonal, Spheroid, Complex, Orientation perception

Types: Habitable, Work, Individual, Group, Play (Individual, Group),
Sleep (Individual, Group), Recuperation, Passage, Storage,
Risky (Work, Play, Passage, Storage)

Ownership: Individual, Shared, Ambiguous

Habitability: Temperature, Air Pressure, Air Composition, Volume, Height, Gravitational

Perception, Connections, Adjacencies

Controls:

Objective: Temperature, Air volume, Humidity, Ambient lighting, Task lighting, Entry, Exit
Dominance: Human, Machine

Groups: Members, Shared Ownership, Shared Beliefs, Shared Capability,
Shared States. Shared Actions

Figure 2.1 — Human-Environment Interaction Issues

In other words, the team collects sufficient data to perform a comprehensive interdisciplinary POE (post occupancy evaluation), as described in Chapter 1. The total volume of data collected would be staggering, of course. In addition, the data is disparate in content, and parts of it would be meaningful to certain scientists in the team but not to others.

Yet, if the team's goal is to use this data to meaningfully inform the design and use of the next-generation MARS, they must devise ways to manage the complexity of the interactions in the data, without losing sight of the important issues, most of which are influenced by multiple inter-disciplinary factors. If the scientists propose to use simulation as a tool to assist the analysis (as is proposed in this thesis), they must devise ways to include all or most of these issues in the simulations, albeit not necessarily all at the same time.

2.1.1 Relevant Modeling Systems

The present research is certainly not the first regarding scenarios where human beings, built environments, and computer technologies interact. The Architecture-Engineering-Construction (AEC) industry's literature includes a substantial section regarding so-called "intelligent" buildings (IB), as discussed in Chapter 1. In this literature for "intelligent" building systems, human beings and everything about them are with distressing frequency relegated to a simple diagrammatic "user" symbol, if not omitted altogether from system diagrams and architecture models. Since authors who work for automated building system and control manufacturers are very well represented in this literature (Caffrey 1988, Pauers 1988, Mool 1989, Heller 1990, Webb 1990, Clapp & Blackmun 1992), perhaps this should be viewed as inevitable in the early decades of a technology-driven phenomenon.

However, even if understandable, the situation is cause for concern as these automated "intelligent" systems become ever more ubiquitous, especially as they become untested controllers of human life. Since the authors referenced above wrote most systematically about IB's in formal terms, a brief examination of how, or if, they set out to model or otherwise investigate these buildings follows.

Pauers (1988) posits that "system integration" means IB's, in the title of the referenced article. He notes how as early as "the 1970's, building fire alarm systems were being tied into the HVAC (heating-ventilation-air conditioning) and fan control systems," and that the U.S.

government General Services Administration (GSA) contributed to the advent of system integration by mandating "automatic control of major building security functions, elevators and lighting." Pauers' article includes six charts or figures which model in rather disjointed fashion what he means by "system integration" or "intelligent" building. In a sidebar he notes that, "There is no intelligence threshold past which a building 'passes or 'fails'. Optimal building intelligence is the matching of solutions to occupant needs." Except for images of office and construction workers in the graphical decoration accompanying the article, there is no further mention or representation of users, occupants or any other human beings, except for the "customer," i.e., the building owner/developer.

Caffrey, who was at one time chair of the now-defunct Intelligent Building Institute (IBI), does a bit better with this. In the center of the large system architecture diagram that he labeled "Characteristics of (IB) elements," there is a one-inch diameter circle, in which are centered the words "Owner and Occupant Needs." The four rectangles, filled with "Systems, Structure, Management and Services" characteristics, radiating in cardinal directions from the circle, have small triangular extensions which all point to the circle. Otherwise, "occupant needs" are mentioned in a few places in the text with no elaboration.

Mool asserts that "systems integration" improves performance. The single system architecture diagram in his article, labeled "Levels of Control" does not include any feature identified with human beings. "Operators" are the only active human beings referred to in the text of the article, while in one instance "occupants" are referred to as the recipients of intercom system broadcasts.

Heller, Webb, Clapp & Blackmun did not attempt any systematic modeling of IB's. However, another batch of engineers did, all of them in Japan (Horitake 1990, Kujuro & Yasuda 1993, Mitsui 1991), or collaborating with Japanese IB researchers (Finley *et al* 1991). Interestingly, several of these engineers assert points regarding users and occupants, and elaborate upon them.

Most assertively of all, Tateishi (1988, 1989) insisted that IB's must "offer a working environment best suited to the individual needs of the worker—the building as 'servant of the people'." In the 1989 article his diagram titled, "Intelligent Building: Component Space and Its

Factors", like Caffrey's, features a large "people" circle, but labeled much more extensively, as illustrated in Figure 2.2.

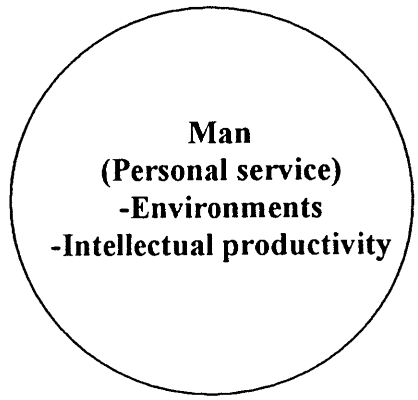


Figure 2.2 – Tateishi's diagrammatic representation of human issues in IB's.

Tateishi goes much further than this in the text, carefully evaluating some relevant occupant issues, as including two tables focusing on workers. These are labeled "Office Floor Area Per Worker (Effective Floor Area), and "Number of Workers Per Office Automation Equipment." Tateishi thus considers human-environment interaction far more than does any other technically oriented author in this literature. It appears, however, that Japan's engineers, while motivated to do so, have yet to devise a method, tool or system that enables systematic examination of "human" issues in IB's. This may explain why researchers and construction firm staff members in Japan strongly encouraged the author to pursue the research presented in this thesis.

2.2 Introduction to Brahms

Brahms is a high-level artificial intelligence (AI) research language and system which permits complex simulations of human beings interacting with technological objects and systems. Brahms has not previously been used to model human and built-environment interactions. One of the important goals of the research discussed here was to determine if Brahms is an even remotely suitable tool for such modeling.

The developers of Brahms explain their goals for the system as follows:

> *Brahms... is a multi-agent simulation framework for modeling work practice, incorporating state-of-the-art methods from artificial intelligence research and insights about work and learning from the social sciences... models consist of groups of agents with context-sensitive, interactive behaviors. Agents are located, mobile, and have knowledge and changing beliefs. Groups may define job functions, teams, people at a certain location, or people with certain knowledge and beliefs...*
> *Brahms... enables modeling activities of people during the day—how people spend their time— emphasizing information processing communication in different modalities (phone, fax voice mail, face-to-face, databases), and location-specific interaction (meetings, chance conversations, teamwork). Thus, Brahms allows modeling a community of practice—a group of people who participate in some shared, choreographed interaction, usually inviting collaboration between individuals with different roles and experience...*
> *Brahms... is primarily designed to be a tool for learning how to design work systems, as well as a means of embodying what we learn over time...*
> *Brahms models... make social processes visible... provide a holistic perspective on how work gets done... may he used to... redesign organizations, facilities, and procedures; design information processing technology... ; develop an instructional system with role... ; develop software ('intelligent') agents in Brahm's ...; provide a researcher workbench for developing models of organization, problem solving and learning in social systems... (*Clancey *et al* 1996*)*

The ambitious goals of the Brahms developers as stated above will be examined more carefully as they relate specifically to human-environment interaction analysis.

2.3 History of Brahms

Brahms was initiated by a team of researchers associated with the former Nynex Inc. Science & Technology Division and the Institute for Research on Learning (IRL)[3] of Palo Alto, CA, USA. It was originally developed "to replace a conventional business process modeling tool and is directly based on Nynex's and IRL's experience in studies of work practice for the purpose of systems design" (Clancey *et al* 1996). Nynex sought a method for reengineering its business operations, especially those calling for new hardware and software systems, which

[3] http://www.irl.org/

would not result in more problems than they were designed to solve, a common fate of corporate reengineering efforts. The company postulated that IRL's social-science-based methods held promise to address complex human issues often overlooked in previous reengineering approaches.[4]

After the initial development effort led to encouraging results, Nynex and IRL filed a patent application[5], which is pending at the time of this writing. However, well before the patent process was complete, Nynex merged with Bell Atlantic Inc. and the Science & Technology division was disbanded. Members of the original team who continue to be involved with the development of Brahms include:

- William Clancey[6], formerly with IRL, then serving as a consultant to Nynex, now lead HCC research scientist at the NASA Ames Research Center[7].
- Maarten Sierhuis[8], formerly with Nynex Science & Technology, now a research scientist at the NASA Ames Research Center.
- Ron van Hoof, formerly with Nynex Science & Technology, now a computer scientist at the NASA Ames Research Center.

Because of these events, Brahms has been an underfunded development project for several years. At the time of this writing, the software applications that are part of the total Brahms system most clearly suffered from the history just described, as they are largely incomplete, an issue that will be detailed in later chapters.

Why Brahms?

The most complete portion of the Brahms system is the computational framework defined in the language specification (van Hoof 1999), and the philosophy or conceptual model imbedded in the specified programming language. This is fortunate, since the philosophy and language are the most important features indicating that Brahms may be an appropriate tool for human-environment interaction modeling and analysis. The hypothesis that it is indeed appropriate is discussed in detail in the next chapter.

[4] Verbal communication from the Brahms development team.
[5] Patent Pending, Attorney Docket Number 07602/003001.
[6] http://home.att.net/~wjclancey/
[7] http://ic.arc.nasa.gov/ic/HCC.html
[8] http://ic.arc.nasa.gov/ic/maartens.html

Page intentionally left blank.

3. Brahms (Business Redesign Agent-Based Holistic Modeling System)

3.1 Brahms as a Modeling/Simulation Software System

To use Brahms, an investigator first creates a Brahms model of the scenario of interests. Such a model is based on the actual behavior of human beings as they interact with artifacts, other human beings and the physical environment of the scenario. The Brahms system was developed in the context of business reengineering or workplace analysis, to capture *work practice*, the actual behavior of people working. The concept of "work-practice" extends similar "work-X" concepts commonly used in business and workplace research, such as "work-process", "-procedure", "-flow", "-tasks".

Work practice includes informal or apparently incidental actions and interactions—such as workers routinely conveying information to each other during coffee breaks— which are usually ignored in most methods of workplace analysis. That such details contribute significantly to the "practical knowledge" constituting the "practice" of any professional or other worker, is one of the central tenets of the intellectual and philosophical discipline out of which the Brahms system arises.[1] To enable capture and analysis of such informal or incidental behaviors along with the formal behaviors typically called "work", Brahms was deliberately designed to be quite general (Sierhuis 1999); it is purported to enable modeling of any conceivable human, animal or artifact behavior.

The Brahms developers consciously incorporated the methods and language of social sciences such as anthropology into the Brahms framework, thus they use the term "ethnography" for the process of collecting data for a scientific investigation of a workplace or other social scenario. While it will not be examined here in detail, ethnography is discussed by Clancey (1999) specifically in relation to the MARS project[2]. In short, ethnography consists of investigation of activities, places and concepts:

- What people do where, when, with whom, and why
- The flow and intricacies (the "choreography") of human interaction

[1] Inference from verbal communication with the Brahms team.
[2] http://home.att.net/~wjclancey/HCC_at_HMP-99_Mars_Society_.pdf

- Understanding total systems, including facilities, artifacts, tools, organizations, people, personal identities, spaces, and locations[3]

Note that ethnography has been traditionally used in observational and descriptive sciences such as anthropology, but its use has been increasing in settings where the primary motivation is change, such as corporate reengineering efforts, or "business anthropology." (Clancey 1999)

3.1.1 Input: a Brahms Model

The process of developing a Brahms model begins just as does any other modeling effort, with gathering data. The model builder must choose the scenario, the boundaries and the constructs of interest, and acquire the data necessary for the investigation. A model—any model—is created to explore an issue, test a hypothesis, or to pursue an objective of some other sort. Whatever the issue or objective, the model builder chooses the important items or features to include, and then must map them to constructs included in the framework in use, here the Brahms framework.

3.1.1.1 Developing a Brahms Model: Data Acquisition

The first action in the process of developing a Brahms model of a workplace scenario may be interviews with each participant in the scenario; the investigator asks each person to describe how they perform their work. The questions asked and the level of detail required in the data collection process will depend upon the overall goals of the investigation. After being encoded, each person's responses can be collected and edited to form a story, which represents the person's description of how s/he performs work.

To develop the required structure of a Brahms model from such stories, the investigator identifies key Brahms constructs in each story. Figure 3.1 (next page) shows, on the left side, a small portion of someone's story of their work. The worker in this case is a repair specialist who works on telephone circuits. The figure represents the investigator's notes after processing data collected in previous interviews. The investigator highlighted words in the story that can be associated with Brahms constructs. For example, "I," "Bill," "my first-liner"—words referring

[3] http://att.home.net/~WJCPublications/Mars99ConfSlides

to persons operating in the scenario—are marked as potential "Agents". People are always modeled as agents in the Brahms system. Non-living physical objects are usually modeled as some type of object, although certain complex physical systems—for example robots—may be modeled as agents. In the English language, nouns are used to represent persons and objects, the latter of which may be a) physical items or b) concepts. Since these two notions are modeled somewhat differently in Brahms, nouns within the story are identified separately as "Agents," "Objects (Things)," and "Objects (Concepts)."

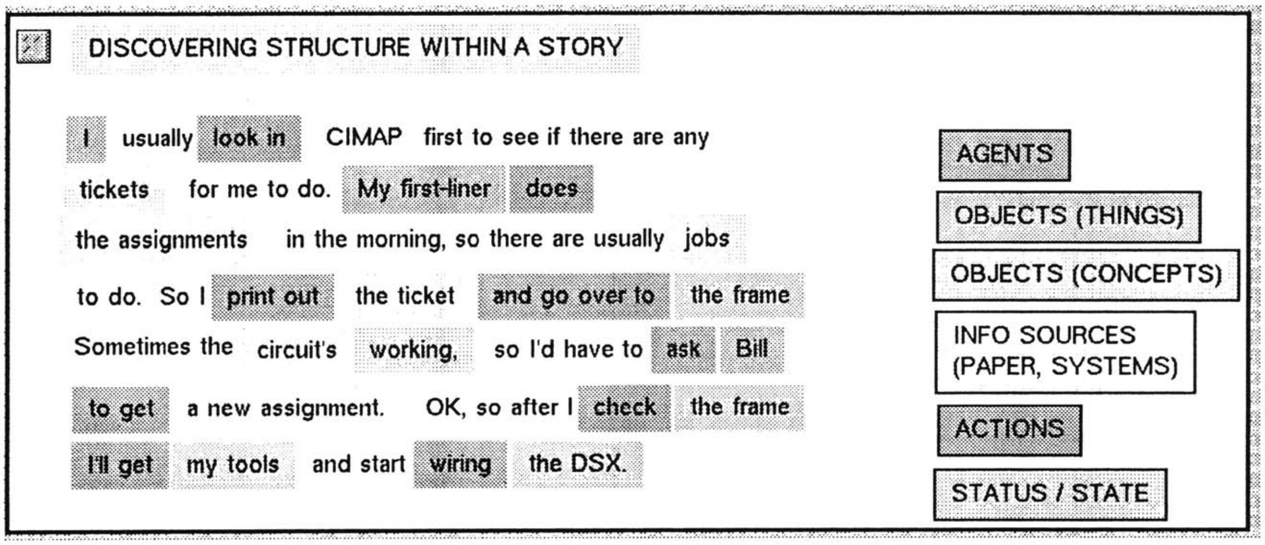


Figure 3.1 – Discovering Structure within a Story[4]

Likewise, the investigator has identified verbs in the story, and associated them with "Actions." Features of the story that indicate sources of information for the participants in the scenario are identified as "Info Sources." Features of the story that indicate the status or states of a system, agent, object or the environment are identified as "Status/State."

3.1.1.2 Acquiring Additional Data for the Developing Model

After identifying the basic Brahms constructs in the story, the investigator then seeks additional information by asking questions. Again, the questions asked and the level of detail required will depend upon the goals of the investigation. Sufficient data must be acquired to enable exploration of the work process or other scenario at a level of detail appropriate to the overall goals. Figure 3.2 (next page) shows on the left, elements of the same story snippet from

Figure 3.1, sorted by the Brahms constructs to which they have already been mapped by the investigator, and on the right, the additional questions deemed appropriate to ask at this point.

An alert reader may notice that the "Actions" construct appearing in Figure 3.1 is not listed in Figure 3.2. This is because in the Brahms system, actions are identified with the formal concepts "Activity" and "Workframe." These constructs essentially "drive" a simulation based on a Brahms model, that is, the a simulation run moves forward in time as a result of how activities and workframes are defined and modeled. Thus the investigator in this case would have gone further to break down the story in the dimension of "Actions", defining them in additional charts, not shown here, similar to Figure 3.2.

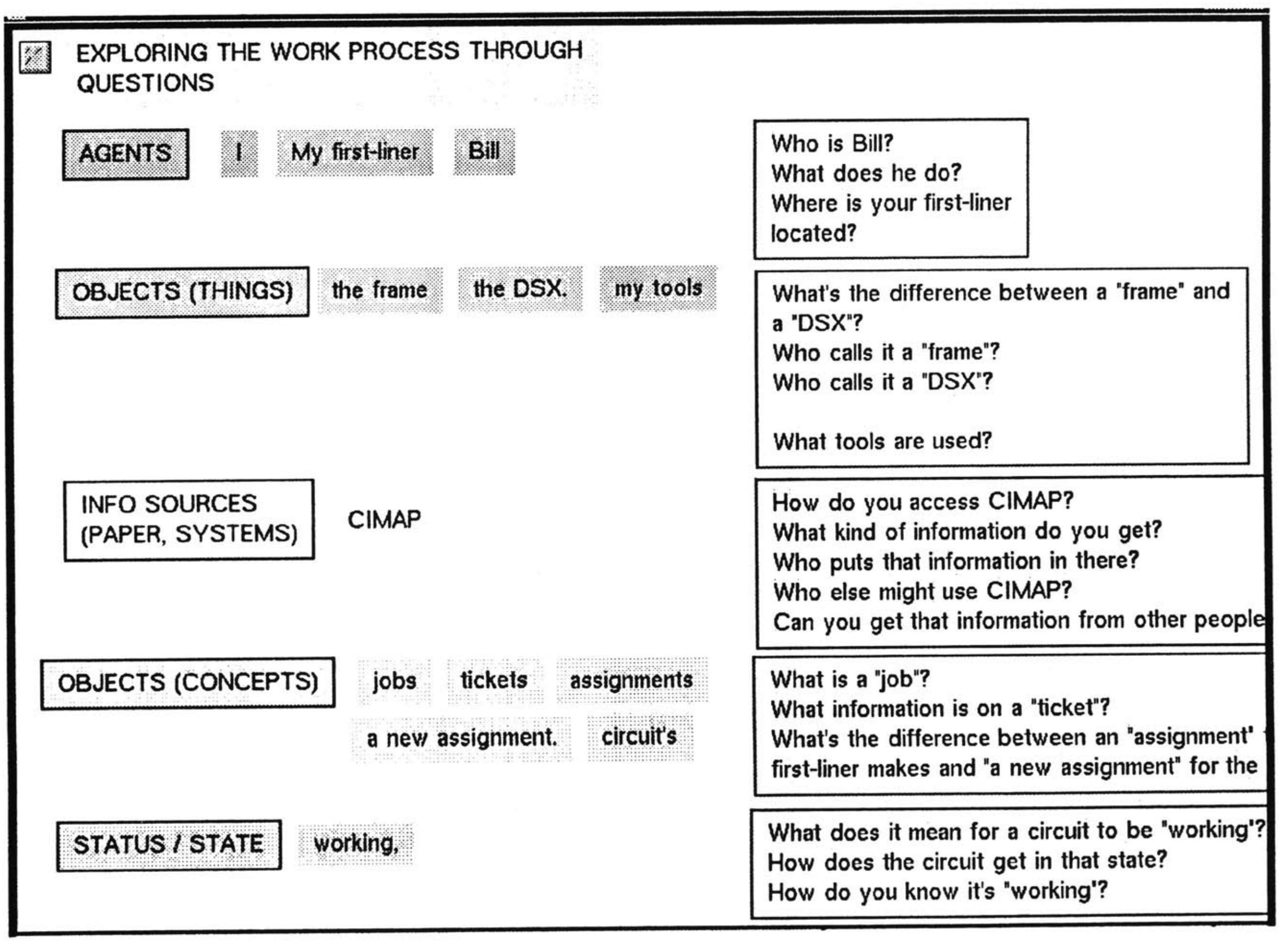


Figure 3.2 - Exploring the Work Process through Questions

[4] Figures 3.1, 3.2, 3.3 taken from imbedded charts in the Brahms G2 Engine. Used with permission.

3.1.1.3 Moving Towards a Model

In Figure 3.3 the investigator has further structured the repair specialist's story snippet, separating and listing the Brahms constructs represented in the story, preparatory to modeling this scenario. This process will be examined in detail in Chapter 4 for the specific case of the MARS project model.

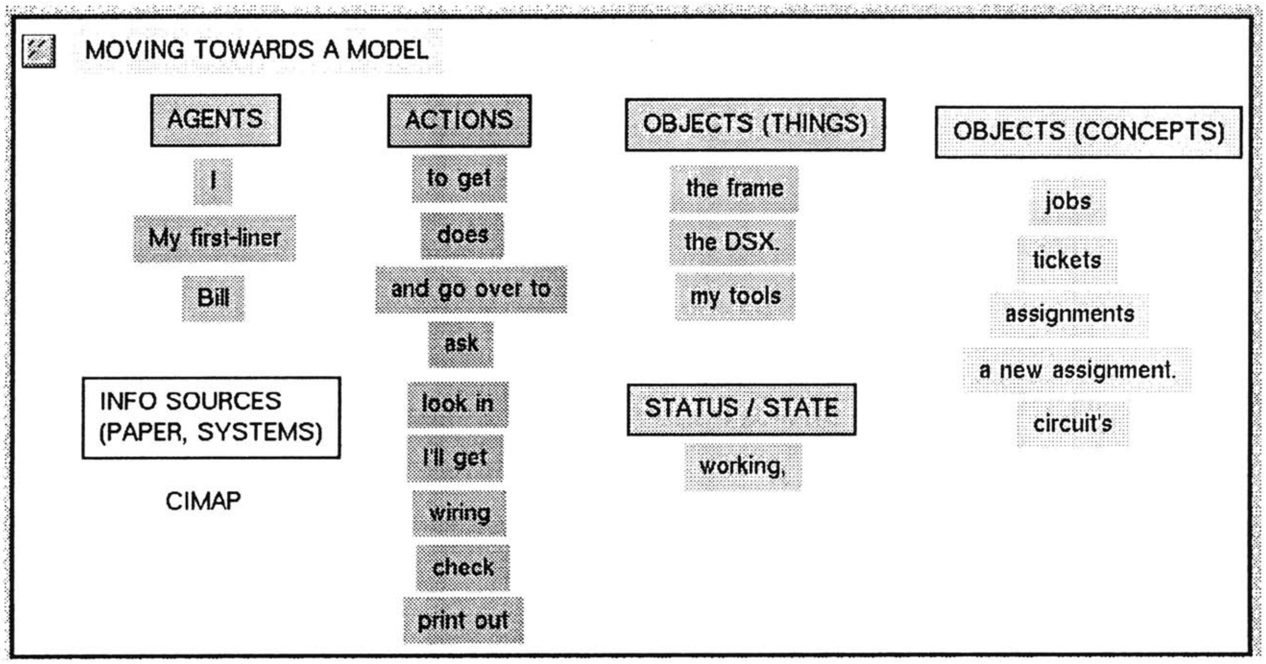


Figure 3.3 – Moving Towards a Model

3.1.1.4 Brahms code

The next stage in developing a Brahms model is to write code in the Brahms language. A "wizard" designed to expedite this process exists in the Brahms software system. It and the other components and processes of the system are examined in detail in further sections of this chapter and the next. Figure 3.4 (next page) shows an example of input code for a Brahms model. In this case, the model is a representation of an aspect of the Space Shuttle.[5] Brahms code looks like this whether written by hand with a text editor or developed with the wizard system. Programmers will notice that this code looks much like C++ or Java code, except for the specialized construct keywords not found in those languages.

[5] Code example transferred with permission from a machine controlled by Maarten Sierhuis.

```
//© 1999 NASA & Maarten Sierhuis.  All rights reserved.

import Brahms.Base.BaseGroup;
import Brahms.Base.BaseClass;

group InterfaceSystem memberof BaseGroup {
    attributes:
        private Building getTemperature;
}

agent SpeechRecognizer memberof InterfaceSystem {
    initial_beliefs:
        (the getTemperature of current = LowerDeck);
        (the getTemperature of current = MidDeck);
        (the getTemperature of current = UpperDeck);

    activities:
        communicate GetTemperature(Building deck, SpaceShuttleRobot robot) {
            max_duration: 5;
            with: robot;
            about:
                send(the getTemperature of current = deck);
            when: end;
        }

    workframes:
        workframe AskForTheTempOfDeck {
            repeat: false;
            variables:
                collectall(Building) X;
            when (knownval(the getTemperature of current = X))
            do {
                GetTemperature(X, PSA);
            }
        }
}
group SpaceShuttleRobot memberof BaseGroup {
    relations:
        private Building communicatedTemperatureOf;

    activities:
        composite_activity GetDeckTemperature(InterfaceSystem Y) {
            activities:
                move GoToDeck(Building deck, int pri) {
                    priority: pri;
                    location: deck;
                }

                primitive_activity DetectTemperature(int maxd, int pri) {
                    priority: pri;
                    max_duration: maxd;
                }
```
(clipped)

Figure 3.4 – Portion of Brahms Sample Code File "SpaceShuttle.b"

The code shown in Figure 3.4, while no more than a snippet, yet exhibits several Brahms language features that will be recognizable by or conceptually familiar to a C++ or Java programmer:

The notion of importing libraries or packages is evident:

```
Import Brahms.Base.BaseGroup;
import Brahms.Base.BaseClass;
```

The general form for specifying the creation of data constructs and the inheritance relationships of such objects are noticeably similar. Where in Java and C++ one may specify a "class" data object and what other class/es it inherits from (or is otherwise related to), in Brahms one may similarly specify data objects, such as a "group," an "agent" and various others not shown in this code example:

```
group InterfaceSystem memberof BaseGroup {
    attributes:
        private Building getTemperature;
}
agent SpeechRecognizer memberof InterfaceSystem {
    initial_beliefs:
        (the getTemperature of current = LowerDeck);
        (the getTemperature of current = MidDeck);
        (the getTemperature of current = UpperDeck);
```

Other portions of this Brahms code snippet resemble Java or C++ "methods", that is, code that specifies the active behavior of data objects:

```
activities:
    communicate GetTemperature(Building deck, SpaceShuttleRobot robot) {
        max_duration: 5;
        with: robot;
        about:
            send(the getTemperature of current = deck);
        when: end;
    }
workframes:
    workframe AskForTheTempOfDeck {
        repeat: false;
        variables:
            collectall(Building) X;
        when (knownval(the getTemperature of current = X))
        do {
            GetTemperature(X, PSA);
```

In the next chapter, the process of moving from the general ideas of a system to be modeled, toward code such as this, will be discussed further.

3.1.2 Components and Process

The current Brahms software system consists of three major components, presented here in the order they must be used to process a model: first the "Brahms Builder", then the "G2 Engine", and finally the "Agent Viewer."

3.1.2.1 Brahms Builder

Brahms Builder is a Java application that compiles code such as shown in Figure 3.4; it "builds" a model from such code. Since the current "Engine" (see section 3.1.2.2, next) requires its input to be written in the proprietary G2 language, in files called "knowledge bases" (with extension ".kb"), the Brahms Builder converts the code representation of the model into G2 ".kb" files. The Builder checks the code for compliance with the Brahms language specification, so that if a code segment compiles, it is likely to run within the G2 Engine environment.

In addition, within the Builder, the "New" command invokes the "Model Wizard" software component. This Wizard, when developed further, should enable someone writing Brahms code to do so more easily than is possible now.

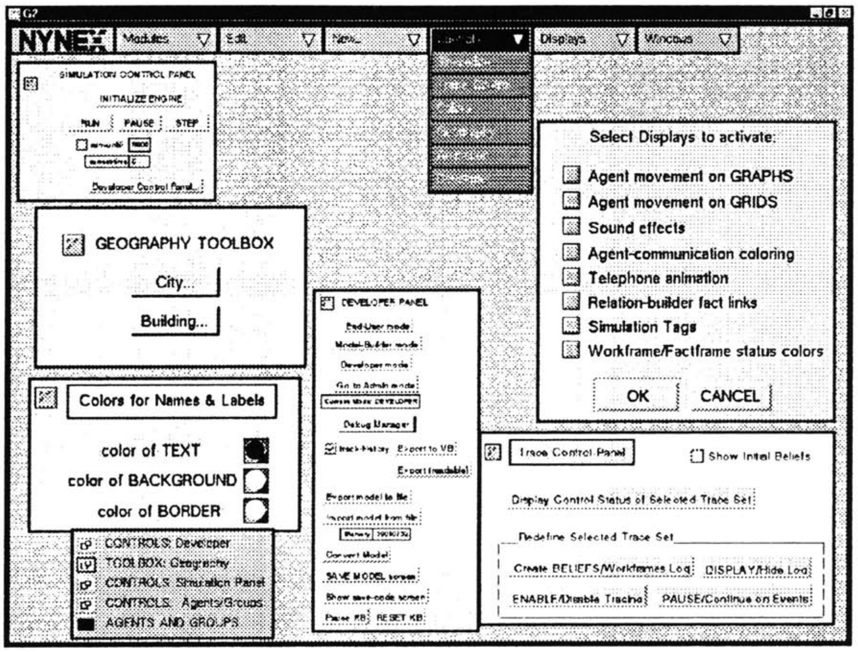
3.1.2.2 The G2 Engine

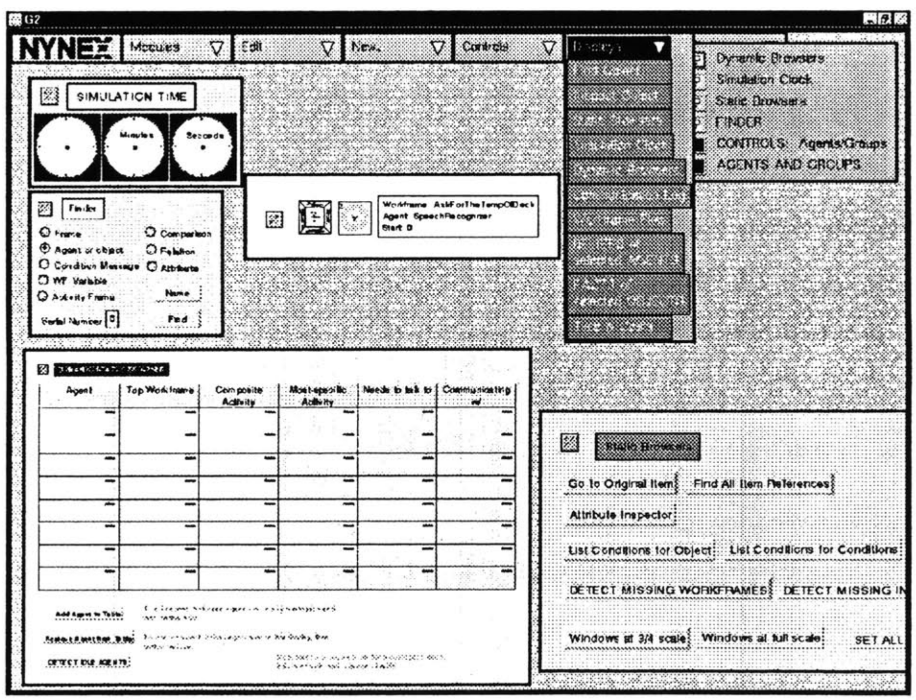
To the Brahms user, the G2 Engine is a "black box." The user follows instructions provided to process a Brahms model within the G2 environment ("a simulation run"). The user then issues commands to export the simulation results. G2 emits a large text file in response. While G2 includes a rich set of features for developing, implementing, modeling, simulating, and displaying complex industrial systems (for example, G2 is used to run entire chemical processing plants, and the scheduling system of the Panama Canal), its use within the Brahms framework is as a simulation engine. G2's robust and well-tested artificial intelligence (AI) features apparently enabled Brahms to be prototyped and implemented quickly, with minimal "reinventing the wheel." However, for reasons that are not clear to the author of this thesis, the Brahms development team considers G2 inappropriate for the stage of development that Brahms has reached.[6] It is worth noting that since G2 is an expensive proprietary product, complete

[6] Maarten Sierhuis verbal communication December 1999.

dependence upon it could inhibit the use of Brahms as a research tool, since any robust research effort is likely to require purchase of multiple G2 licenses.

Figures 3.5 and 3.6 are screen shots of the Engine running in G2. Very few of the windows and menus visible in these views are necessary to perform a simulation run.





Figures 3.5 and 3.6 – Brahms within G2

3.1.2.3 Agent Viewer

Agent Viewer is the final component in the current tripartite Brahms system. Like Brahms Builder, it is a Java application, the input for which is the text file created by the G2 Engine. The user first issues a command to "create a new model" within Agent Viewer, specifying location of the input text file. This creates an empty Microsoft Access database, which Agent Viewer then automatically populates with data from the text file ("parsing the file"). Once this process is complete, the user may use Agent Viewer to examine the agents, objects and other constructs defined in the model, and the interactions occurring among them in the simulation run.

3.1.3 Output

Brahms output consists of, a) visualizations in Agent Viewer and, b) the Microsoft Access database created in Agent Viewer by parsing the G2 text output.

3.1.3.1 Agent Viewer as Output

Figure 3.7 (next page) shows Agent Viewer in use to "visualize" the output of a Brahms simulation run. The left windowpane shows the Brahms language constructs in the model, listed hierarchically. The user may select agents and other constructs from the left windowpane, with the result that a time-line visualization of the selected agents and objects appears in the right windowpane. Vertical lines with terminal dots show connections and interactions between agents, objects and other constructs. The right windowpane is interactive: selecting one of the displayed text fields, lines, dots and other displayed items, reveals popup explanations, menus, text entry boxes, options, or additional windows. Menu items permit the user to examine the time line at several scales, from full-day intervals to 5-second intervals.

Through these interactive features, Agent Viewer allows an investigator to examine *all* of the output data of a Brahms simulation run as recorded in the database. Agent Viewer is very flexible, enabling "drilling down" into the data to any level of detail, using one or more of the interactive features. It is thus best used interactively. Capturing all of its features with screen shots while examining anything more than an extremely simple Brahms model would require dozens of figures similar to Figure 3.5.

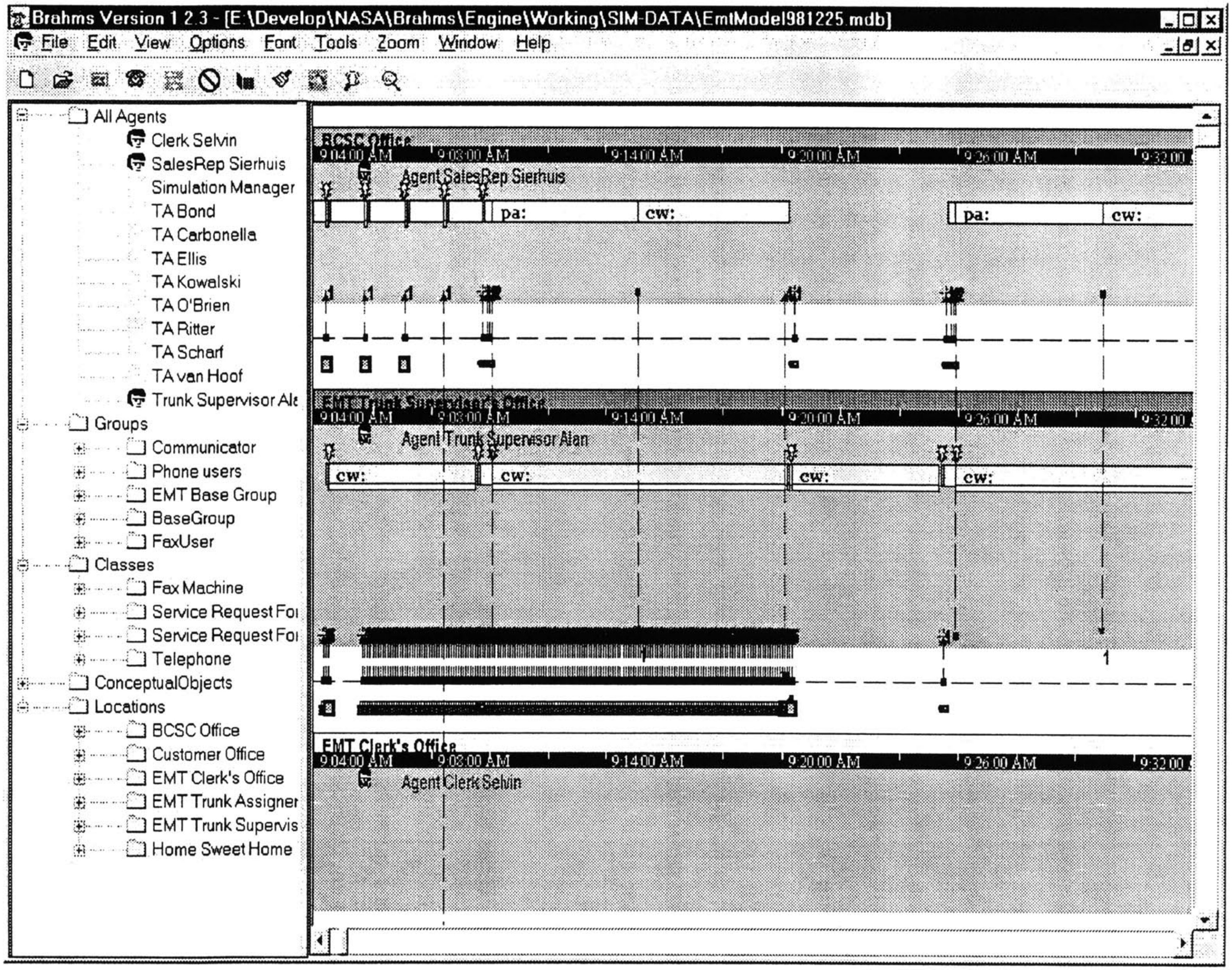


Figure 3.5 – Agent Viewer

3.1.3.2 The Brahms Output Database

The Brahms output database consists of 41 data tables and 43 queries, populated by the events occurring during the simulation run. The database is the repository of a large amount of data, transformed from its raw state into a highly structured format. A complete analysis of the database, which is fully relational, will not be included here in part because of the incomplete patent application process.

The Brahms system shows promise of accomplishing at least one of the goals of human-environment interaction analysis noted in Chapter 2: managing large amounts of complex and disparate data, intersecting portions of which are of interest to investigators from multiple disciplines.

Since the Brahms Agent Viewer application creates output visualizations entirely from the database generated by the Engine, it is conceivable that the Brahms system may be extended by using the same database for other purposes. This possibility will be discussed further in a later chapter.

3.1.4 Uses of Brahms Output

Since any investigation seeks to answer questions of some sort, it is useful to conceive of at least one question that Brahms output may help answer. There are many candidates in the human-environment interaction arena; for simplicity's sake, one of the human factors discussed in Chapter 1 will serve to illustrate the point: *stress*.

Question: What features of a habitable space affect human beings differently in ordinary "day to day" circumstances, from how the same space affects human beings in extraordinarily stressful (i.e. life-threatening) circumstances?

Brahms output value: First, assume that all of the important factors of both human beings and the habitable space were successfully modeled in Brahms. This assumption will be tested and discussed later in this thesis. Agent Viewer and the simulation data it visualizes could conceivably allow true experiments to enable the determination of answers to the question. It is difficult to imagine how one could conduct *true* experiments to investigate this issue with human subjects. True experiments require random assignment to treatment and control groups, careful control of all known variables (often resulting in "double blind" procedures, etc.), statistically significant numbers of subjects in each group, and so on. The simple matter of ethics precludes crucial aspects of true experimentation. Brahms output—if the language and its software implementations were tested and validated—could fill a gap in this field by enabling research that is largely impossible now. It is not difficult to envision a Brahms model of a habitation unit run with varying human stress conditions, controlling for all other appropriate variables, and then examining these different runs in the Agent Viewer or another application that uses the output database.

3.2 Brahms as a Modeling/Simulation Language

3.2.1 Current Status vs. Anticipated Developments

As noted in Chapter 1, the most complete portion of the Brahms system is the computational framework defined in the language specification (van Hoof 1999), and the modeling philosophy imbedded in the specified programming language. The language is still evolving, and new specifications are published frequently. Certain desirable modeling capabilities consistent with the philosophy are planned[7] but not implemented in the version of language specification used here. Only the anticipated features that are especially relevant to human-environment interaction will be discussed in the sections and chapters to follow.

3.2.2 Brahms Philosophy

The philosophical framework in which Brahms was developed specifically intends to perform modeling within "a theory of human social systems." (Sierhuis 1999) A clear distinction—which the developers present as crucial—is drawn between animate "intentional" objects and inanimate "unintentional" objects. Even though physical objects and systems in the real world can at times, (if they are programmed to appear so), "act" as though they have intentions, the Brahms philosophical stance is that only living systems such as human beings and animals are intentional. The developers note that while Brahms is an Agent-Oriented Language (AOL) rooted in the artificial intelligence (AI) tradition of "strong agency", most other AOL

[7] Development intentions were conveyed to the author verbally by several members of the Brahms team.

systems define all strong agents as intentional agents. (Sierhuis 1999) Since in Brahms the concept of "strong/intentional agent" is reserved for animate agents, another concept is required for inanimate objects. In Brahms, the formal concept of "Object" is used for all inanimate objects or systems, even if such systems can mimic the intentionality of animate objects such as human beings. Note that when coding a Brahms model one may choose to define a robot, for example, as an "Agent" rather than an "Object", the intention of the developers is that this would be a special circumstance used only when clearly warranted.

3.2.3 How the Philosophy is Implemented in the Language

3.2.3.1 Agents

As noted previously, people are always modeled as agents in the Brahms system. Agents within the system can "do" most of what actual human beings can do. Examples follow:

- Act – modeled as "activities", "action"
- Know – modeled as "beliefs" (which can change dynamically)
- Think – "thoughtframes", "conclude"
- Detect facts – "detect", "detectables"
- Perform routines – "workframes"
- Associate in various ways – "groups", "relations"
- Interact with physical items – "object"
- Wear or posses things – "contain"
- Conceive – "conceptual_class" and "conceptual_object"
- Move – activity "move"
- Be somewhere – "area" (world, city, building, etc.), "location"
- Go somewhere – "path", "destination", "distance"
- Communicate – "communicate", "broadcast"
- Get paid – "cost"

- Take time to do things – "time", "max_duration", "min_duration"
- Set priorities – "priority"
- Create – "create_object"

The list above is not meant to be exhaustive, but illustrative. The concept of an "Agent", representing human beings, is defined as "the most central element in a Brahms model." (Sierhuis 1999) The daily activities of actual individuals may be modeled, as well as the activities of abstracted groups of individuals, this is the "Group" concept. The participation or membership of agents within various types and hierarchies of groups can change dynamically.

3.2.3.2 Objects

As noted earlier, in the Brahms system inanimate things are expected to be modeled as some type of object. Objects can "do" most of what agents can do, including possessing "thoughtframes," which are representations of inference processes, even though one might suppose that objects can not think. The most notable difference is that objects can be part or members of "conceptual objects", which represent the notion that people often conceive of a group of independent objects as part of a single "thing." For example, "an order" in many business settings can not be represented by a single artifact, while what constitutes an order changes frequently. During work hours an order may be represented by a single piece of paper one moment, but by two pieces of paper the next moment (i.e. one worker faxes "the order" to another), and by a record in a database in the moment after that. At night or over the weekend when the staff is not working, an order may be represented by several pieces of paper or regions of magnetic media located on several different desks, not necessarily in the same building or even on the same continent. An order could even exist as a belief entirely within the mind of one or more workers.

In addition to this notion, objects differ from agents in that they can be defined as resources, but agents are not described in this way.

3.2.4 Differences from other Systems

Brahms is similar in some ways to other programming, modeling or simulation systems, but it also differs from the other well-known systems. These differences are discussed briefly below.

- Object-Oriented Programming (OOP): Brahms incorporates many of the concepts used in OOP, such as objects and object instances, classes, inheritance and multiple inheritance, abstraction, attributes, association, aggregation, constraints. (Rumbaugh *et al* 1991) Indeed, a Brahms model builder who does not possess a thorough grasp of OOP concepts and techniques will have difficulty learning or using the Brahms language. However, in C++ and other general-purpose OOP languages, notions or constructs such as "agent" are not included in the language, nor are the capabilities and techniques of AI built into implementations of the language. For example, if a C++ programmer wished to invoke inference methods such as forward-chaining and backward-chaining, or neural network methods, in the system under development, s/he must include code for each of these techniques in the material submitted to the C++ compiler. In Brahms, these computational techniques are an inherent part of the "Engine" so that the model builder need not be concerned with directly implementing these well-established software technologies.

- Business Process Modeling (BPM): Brahms incorporates or can be readily used to model concepts with a functional perspective such as typically found in BPM, including organizations, roles, product flow, orders, schedules, processes, etc. Typically in these systems the notion of "work practice" is not acknowledged or implemented; the notions of ambiguity, change, exceptions, redundancy and improvisation are de-emphasized or ignored; and social interaction is over-simplified. (Clancey *et al* 1996, 1997, 1998) Brahms goes far beyond these systems by providing a general method for modeling many kinds of social systems and interactions.

- Cognitive Process Models (CPM): Brahms incorporates the knowledge perspective of CPM, including flow and storage of information, error detection, and problem solving, but does not attempt to model reasoning or calculation internal to an agent or object. Inferences are modeled only to the extent required to represent agent or object behavioral triggers. Thus Brahms models are "not as detailed as models of cognitive skills." (Clancey *et al* 1996, 1997, 1998)

- Architectural-Engineering-Construction (AEC) Modeling: Architects and other professionals concerned with the design of built environments are usually most familiar with models developed with computer aided drawing (CAD) systems, or 3D graphical "rendering" systems. These models represent the physical form, appearance and construction methods of buildings and other structures. The modeling tools typically used by engineers for design of the built environment, such as finite element analysis (FEA), or those used by planners and managers of the built environment, such as computer aided facility management (CAFM), similarly focus on technical aspects important in those fields. Clearly, while Brahms models or simulations could be visualized or animated within the context of these systems, a Brahms model is designed to investigate behavior and interactions, especially those involving human beings, rather than form or strictly physical attributes such as strength or cost.

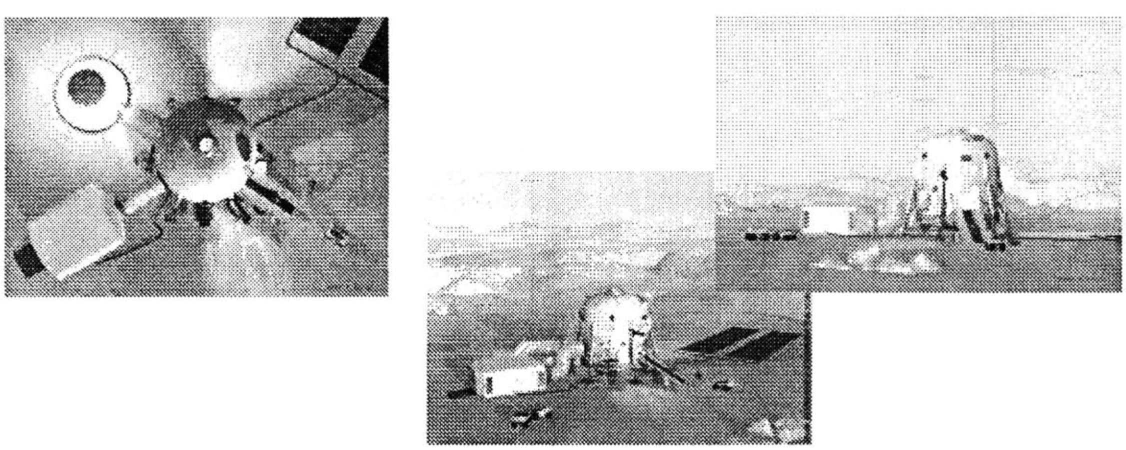
This page intentionally left blank.

4. Modeling the MARS in Brahms

Scientific exploration and research are expected to provide new knowledge about new systems, yet they very often also provide new knowledge about familiar systems. The thrust to explore the planet Mars exemplifies this. Mars exploration will provide new knowledge about a new system, since it will probably be the first planet beyond the Earth that humanity will visit. (Zubrin 1996) Mars exploration will also provide new knowledge about a familiar system, the Earth itself. In similar manner, the scheduled MARS habitation project in Haughton Crater will provide new knowledge about the habitation unit itself and its fitness for its intended purpose. The author expects that it will also provide knowledge about familiar systems, that is, complex life-support environments that serve vulnerable populations on Earth, such as hospitals. This is a significant motivation for the research discussed here.

At the time of this writing, NASA and the Mars Society have not completed the full set of specifications, construction drawings and other design documents for the MARS habitation unit. However, preliminary descriptions and renderings of its layout and appearance are available on the Mars Society web site, as shown below. (Mars Society 1999) These renderings and descriptions, along with Zubrin's book (1996) and documents provided by NASA (1999), were used to guide the development of scenarios—stories—that were further developed into the Brahms models discussed in the remainder of this thesis.

4.1 The Mars Arctic Research Station (MARS)



Figures 4.1, 4.2, 4.3 – renderings of the MARS facility (Mars Society 1999)

MARS Technical Specs
September 12, 1999.
The core element of the MARS Project is the habitat unit. Measuring approximately 8.4 metres (27ft) in diameter, the habitat unit will provide 2 floors of living space for up to 6 people at a time. The habitat is designed as a multi-function facility incorporating living and sleeping quarters, work spaces, clean-room laboratories, an exercise area, galley and a sick bay.

The two decks of the unit will be linked by a central shaft and ladder, which will also connect to the main airlock in the lower deck. The lower deck of the unit will also provide storage facilities.

Supporting the main habitat unit will be an inflatable greenhouse and a garage / workshop for storing ATVs and rovers. Both the greenhouse and the garage will be linked to the habitat unit by airlock tunnels. Power will be supplied by solar panels arranged a short distance from the habitat unit. This in itself will be an interesting test - the panels will receive less sunlight than they would if they were situated on the Martian equator, the theory being that if they can provide sufficient power for operations in the Arctic, they will be suitable for use on Mars. (Mars Society 1999)

4.2 Modeling Scenario Assumptions

The following assumptions further guided scenario development:

- Location is a habitation unit like that described above, commonly called a hab or the Hab, somewhere on the planet Mars.
- Time is local planetary time: One Mars hour = 1.03 Earth hour.
- The crew consists of four persons: two scientists, and two engineers (Zubrin 1996).

The following names will represent these four:

Scientist one: "Geo", a geologist. (Name pronounced "joe")

Scientist two: "Bi", a biogeochemist. (Name pronounced "bee")

Engineer one: "Eug", a pilot and mechanic. (Name pronounced "yooj")

Engineer two: "Skip", a mechanic and mission commander

For purposes of this scenario development, Geo and Skip are female, Bi and Eug are male.

4.3 Human/Environment Interaction Scenario: Routine Day

Skip relates:

0700: *My alarm clock wakes me up. I hit snooze once to catch forty more winks, as usual. After the first buzz, I know that Eug is now also awake. Geo may also have been awakened by the soft buzz of the clock but Bi probably was not. We all have alarm clocks built into our bunks, but I am the only one who uses mine routinely. The Hab is so small, my clock usually serves for all who can normally awaken with a quiet sound rather than a loud blast. Bi sleeps later because he needs a loud blast or enough time for his own sleep patterns to wake him. On a routine day such as we are expecting, Bi can sleep until his body or his experiments demand his attention.*

Before falling asleep for my last forty winks, I hear Eug climbing out of his bunk as usual. He will be outside by the time the second buzz wakes me for good. Geo may be up and around by the time I climb out of my bunk, or she may be sleeping a bit longer. All depends on what yesterday was like, and how we set our alarms. Geo uses her voice alarm system more often than the rest of us, programming reminders that are spoken to her as she awakens, so softly that the rest of us can not hear them. She does not need a routine daily schedule, as I do, to feel productive. Her days thus appear to have no routine, but I know she uses the technologies built into the Hab to develop a personal rhythm for each day that enhances her overall productivity. I use the same technologies to enhance the sense of a habitual routine, because that enhances my overall personal productivity. I know that when we look at detailed post-event re-creations of our respective statistically "normal" days, there is actually very little difference between how routine or varying hers or mine are. But we each feel more comfortable, since we each feel that we, not the Hab or Mission Control, are in control of our lives.

I prefer to "take my shower," which is really a high-tech mist-bath system, at night before going to sleep. Upon awakening, I change in my bunk from light nightclothes into in-Hab day clothes, and only then go to "John's bunk", the Hab bathroom. Eug does his personal care routine early every morning while everyone else is in bunk. Geo does hers a little later, while Bi does his at night even later than I do.

0708: *Second alarm, snooze over. I change, head for the bathroom for body functions and toothbrushing. The bathroom recognizes me when I enter, because I wear an electronic ID, as* *all human extra-Earth explorers must, ever since the very first human trip to Mars. At first I wore it imbedded in a pendant on a chain fixed around my neck, but now prefer to wear it imbedded in a nosering in my left nostril. Geo wears it imbedded in an earring; Eug imbedded in an old-fashioned GI-style "dog-tag" necklace; we do not know where Bi wears it, and prefer not to ask. We do know that if he took it off, not only would it be immediately painful, as the tools to remove it painlessly are only found on Earth, the Hab and rover computers would all "have a cow", i.e. alarms would sound and flash.*

As I enter, the bathroom:

- *locks all power toothbrush heads and towels but mine;*
- *adjusts to my preferences: water temperature, drying air temperature, flow rates, mist density and scent for lavatory, mist-shower and toilet;*
- *sets toothpaste flouride content and powered toothbrush speed to those the mission dentist on Earth prescribed;*
- *sets up the urine, stool, and blood tests that the mission doctor on Earth ordered to be run today;*
- *orders me via a blinking light to stick my finger in a small hole for a tiny prick and blood draw;*
- *plays the music I like at the moment for bathroom music;*
- *accepts my dirty clothes into a bin that will wash and dry them, then place them in my labeled drawer outside the bathroom for later retrieval.*

When I leave the bathroom, it automatically sprays clean all interior surfaces, dries itself, and resets back to its ready mode. When are we going to get bathrooms like this back on Earth, I wonder?

***0715**: Refreshed, I look out the nearest porthole to check if anything looks different out there today. Nope, still Mars, red dust and rocks, seventh heaven for Geo and all other geologists. Eug is in the lower level, I can hear him doing the life-support systems and equipment check that he does a couple of times a day at least. I head for my favorite place to start the day's activities,* *the flight deck, now the in-Hab mission command center, conference room, place for many ad-hoc activities, acknowledged by all as primarily "my" office.*

Interruption!

It is clear from the first fifteen minutes of the scenario development above, that if the complete story of a full day—even a routine day— in a combined workplace and habitation unit used by four people were described at this level of detail, the story alone would fill a book the size of a rather lengthy novel. Yet Brahms can apparently be used to model every aspect of the story developed thus far. Rather than continue with the story, it is advantageous to skip to the next stage in developing a Brahms model from a story.

4.3.1 Discovering Structure within the 15-Minute Story

The 15-minute story above is presented again in Figure 4.4 (next page), analyzed and broken down into the following general Brahms constructs: agents, objects (things), objects (concepts), information sources, actions and status/state. Note that an important Brahms construct not mentioned in the diagrams in Chapter 3 is *area*. Brahms does include "area definitions" and "area," enabling description and modeling of cities, buildings, rooms and other spaces. Since such spatial forms and issues are central to human-environment interaction, *area* is included in Figure 4.4. Note that certain items appear to be appropriately described by more than one Brahms construct. A "bunk," for example, may be considered as both an object and an area. At this stage, such items will be duplicated in each of the applicable categories. Brahms enables constructs to be related to or contain each other, thus the working assumption at this stage is that setting up such relations or associations can resolve ambiguities or duplications in the model under development.

It is clear from Figure 4.4 that the detailed story of a routine 15-minute period, even in a rather simple work environment, can be broken down into myriad instances of constructs available in the Brahms language. Obviously, this could easily lead to a sort of "explosion": greatly increasing computational effort for diminishing results. To prevent this, the Brahms model builder must sort the raw list of constructs imbedded in the scenario of interest, using some criteria of fitness for the problem space or hypothesis.

Agents, Groups:	"Geo" (female); "Bi" (male); "Eug" (male); "Skip" (female); Mission Control, doctor, dentist, explorer, human, geologist, biogeochemist, pilot, mechanic, mission commander, rest of us
Objects (Things):	Habitat, alarm, clock, bunk, human body, experiment, voice alarm system, technologies, shower, clothing, bathroom, nightclothes, dayclothes, electronic ID, pendant, nosering, earring, necklace, pendant, chain, toothbrush, toothpaste, towel, toilet, lavatory, computer, rover, dust, rock, light, finger, hole, blood, urine, feces, bin, drawer, porthole, surface, life-support system, tool, label
Objects (Concepts):	"forty winks", sound, snooze, sleep, pattern, routine, outside, day, today, yesterday, tomorrow, reminders, schedule, productivity, re-creations, personal rhythm, habitual, comfortable, in control, personal care, everyone else, activity, life-support system, medical test, experiment, temperature, flow rate, density, scent, music, ownership, preference, expectation, attention, interior, exterior, trip, automatic, manual
Information Sources:	sounds, prior knowledge, reminders, speech, routine, schedule, "detailed post-event re-creations", electronic ID, blinking light, porthole, label
Actions:	sleep, awaken, hit, climb, move, use, program, hear, speak, look, enhance, feel, change, do, go, recognize, care, enter, wear, sound, flash, lock, order, accept, play, sets up, adjust, spray, wash, clean, dry, reset, get, wonder, draw blood, check, test, start, leave, remove, serve, demand, set, look, know, take, function, fix, find, light, stick, place
Status/State:	asleep, awake, comfortable, in-control, night, day, morning, early, late, in-bunk, small, large, completed, wearing, in pain, clean, ready, dirty, clean, normal, powered, warm, hot, cold, refreshed, stale, imbedded, in, out, soft, loud, off, on, automatic, manual
Area:	Mars, Earth, Mission Control, Habitat, bunk, bathroom, place, office, command center, conference room, flight deck, outside

Figure 4.4 – General Brahms Constructs in the 15-Minute Story

As noted in Chapter 3, a model—any model—is created to explore an issue, test a hypothesis, or to pursue an objective of some other sort. Whatever the issue or objective, the model builder must choose the important constructs to *include*, at risk of producing a meaningless model, and also must choose the constructs to *exclude*, at risk of model explosion.

4.3.2 Moving Towards a Model

The hypothesis and construct criterion of fitness for the Brahms model presented in this thesis are as follows.

4.3.2.1 Hypothesis

Part 1:

A Brahms model of human beings interacting with a complex habitable environment encourages the developer to model human beings as powerful, active, intelligent agents rather than passive participants in the processes and activities involved in the modeling effort. It was noted in Chapter 2 that in the literature for "intelligent" building and systems, human beings and everything about them are with distressing frequency relegated to a simple diagrammatic "user" symbol, if not omitted altogether from the model development process. This suggests a method for testing Part 1 of the hypothesis: If this part of the hypothesis is supported, developing a model of a built environment that includes any human presence, by defining the human presence in the model with a single diagrammatic "user" symbol, should be very difficult or impossible in the Brahms system.

Part 2:

A Brahms model of human beings interacting with a habitable environment enables the developer to represent complexities found in real-world human-environment interaction scenarios. A method for testing this part of the hypothesis: If this part of the hypothesis is supported, a Brahms model of a built environment will represent human behavior at multiple levels of detail; the functioning of systems, machines and objects at multiple levels of detail; the functioning of cities, buildings, rooms, areas, spaces and paths at multiple levels of detail; and multiple interactions between all of the items just specified.

4.3.2.2 Criteria of Fitness for Story Constructs

Since time and space constraints do not permit modeling all of the items listed in Figure 4.4, certain items were chosen. The selection criterion follows:

Each construct must contribute to the specific objective of human-environment interaction modeling. For example, the "electronic ID" was considered relevant and thus mapped to the Brahms concept "Objects (physical)", since the story presumes that it provides a communication link between its wearer and various built-in systems of the habitation unit. Whether the ID object is imbedded in a necklace, nosering, or earring was considered irrelevant to the human-environment interaction modeling of this scenario. In like manner, each Brahms construct from the story selected for the model had to meet this criterion.

Note that the items listed in the left column of Figure 4.4 are "informal" high-level Brahms constructs only. They are not formal concepts in the language, which includes many more constructs than shown in the figure. Additional formal Brahms concepts were necessary to realize a complete model. Van Hoof (1999) provides the formal concepts and features of the language.

4.4 The MARS Project Brahms Model Code

A portion of the code for the Brahms model developed for this thesis is presented in Appendix B. Each main model file is named for and defines a formal Brahms concept or a related set of concepts.

The following main files constitute the model:

`MarsModel.b` – This file defines the overarching structure of the model, similar to the "Main" function in a C program.

`MarsGeography.b` – All of the area and path specifications for this model.

`Agents.b` – Definitions of the Agents in the model.

`Groups.b` – Definitions of Groups (of agents) in the model.

`Objects.b` – Definitions of Objects and Conceptual Objects in the model. (Additional files exist named "`Objects2.b`", etc.)

`Activities.b` – Definitions of Activities which are not specified within the specific context of agents, groups or objects.

These files are concatenated into a single document for publication here, but maintained as separate files for compilation and simulation runs. Not all of the code developed as part of this investigation is included in the Appendix; the reason for the omission will be discussed in the next chapter.

5. Analysis, Conclusions and Recommendations

5.1 Using Brahms as a Human-Environment Modeling/Simulation Conceptual Framework and Language

As a conceptual framework for the type of analytic research referred to in this thesis, Brahms is a superbly promising tool.

Part 1 of the hypothesis presented in Chapter 4 is definitely supported: it would indeed be very difficult to develop a Brahms model of a built environment scenario in which all human beings and everything about them are represented by no more than a simple symbol labeled "user." It would be *possible* to develop such a model. One could certainly model objects of all types in a workplace or other scenario, invoking some sort of ghost-like non-agent entities that interact with the objects emphasized in the model. Since Brahms provides a rich toolset for modeling human behavior, yet sidesteps the "explosion" difficulties encountered when attempting to model the complex internal cognitive processes of human beings, a ready critique of such a modeling effort would be: "Why?" What would be the point of developing such a model? If human involvement in the system of interest is not expected, some other modeling and simulation tool is likely to be more appropriate than Brahms.

Brahms does not cripple modeling of well-developed "adaptive" or "intelligent" systems or buildings to provide this robust ability to model human behavior. The investigator of an interaction scenario who uses Brahms to assist in the investigation has access to the same rich toolset to model intelligent systems of all types, including inanimate objects such as computers. In addition, such systems, modeled as objects, can be aggregated to multiple levels of complexity through use of the formally defined "conceptual object" and "conceptual class" concepts. Thus an "intelligent room" as currently investigated at MIT by Coen (1997, 1998, 1999) or a "smart house" (Larson 1999), could be readily modeled in Brahms as the complex, adaptive, feature-rich systems they are. In addition, however, Brahms provides a formal language that enables analysis and testing of those environments with human beings included as significant agents in the scenarios portrayed.

Brahms shows further promise for human-environment interaction investigators specifically. Spatial issues are often severely neglected in modeling systems that are not specifically developed for architects, interior designers or facility managers. Even social scientists often neglect spatial issues and other aspects of real built environments. To professionals in the AEC industry, it is well understood that non-professionals take their physical surroundings so much for granted, that they scarcely "see" any of its features. For example, many people would be hard-pressed to correctly and precisely identify the color of the countertops in their kitchens at home, or the material used for the baseboards in their office.

Brahms is not without some shortcomings in this regard, these will be discussed in the section on recommendations.

In essence, Brahms, like many significant scientific tools developed in the past, shows promise for making the invisible, visible. In this case, the invisible, the variables of the world that we do not know how to "see" very well, is human behavior, social interaction and human interaction with artifacts and the environment. Brahms, as a conceptual framework and modeling language, seems to make a significant step toward changing this.

5.2 Using Brahms as a Modeling/Simulation Software System

5.2.1 Current Status vs. Anticipated Developments

It is important to consider the implications of the history provided in Chapter 2, and the fact that Brahms is both a software system designed as a tool for research; and a research project itself. As a research project, Brahms is the subject of a forthcoming Ph.D. dissertation by Maarten Sierhuis[1], co-developer and project scientist at the NASA Ames Research Center. As a software system, Brahms is still under development, with the current development effort limited by various issues to a very small team, consisting of Sierhuis, Ron Van Hoof, and Bharathi Raghavan, all of NASA.

---

[1] Three unpublished chapters were provided to the author.

The most noticeable "feature" for the first-time user of the current Brahms software system is its incomplete state. The current implementation consists of three pieces, as previously discussed:

- "Builder", which serves as something like a precompiler or translator for the "engine",
- The G2 "engine", soon to be replaced by a Java engine. As noted before, G2 is a commercial AI system more often used for large-scale, real-time industrial control and modeling systems.
- Agent Viewer, another Java application, which provides translation and visualization of the output.

A Brahms model is thus converted from one language to another several times in its journey from input to output.

The only engine available at the time of this writing, the G2 engine, does not support all of the features and concepts defined in the current language specification. The Brahms team states[2] that a new implementation of the engine in the Java development environment is expected within one month of the date of this writing. For this reason, there are no plans to improve or enhance the current G2 engine.

These facts mean that at present:

- There is little documentation other than the language specification (van Hoof 1999), and no user manual. The most complete instructions for traversing the complete Brahms system from input to output were transmitted to the author of this thesis in a single email; evidently, no other start-to-finish document exists. Portions of Sierhuis' dissertation-in-progress are useful for understanding the details of the Brahms system and its concepts, but this document understandably has restricted circulation. Since the G2 engine is expected to be replaced quite soon, there is understandably little motivation to document the exact deficiencies of the current engine with respect to the language specification.

---

[2] Verbal communication.

- Little has been publicly published about Brahms, primarily because the patent process had not advanced sufficiently until quite recently. One journal-published paper is (Clancey *et al* 1999), and additional miscellaneous unpublished papers can be downloaded from several web sites.
- The first "user" of Brahms was Maarten Sierhuis. The author of this thesis was the *second*. Libraries, samples and examples are more or less the same items, all the work of Sierhuis.

As an inevitable consequence of all this, at present Brahms would score quite low on any reasonable scale for "usability," "user-friendliness" and the like.

5.2.2 The Current Brahms Experience

The author's experience with development of a Brahms model and attempts to run it through the complete system is briefly documented in this section. Following that, recommendations for the Brahms development process are presented.

For a beginning user of any complex software system, the first problem is usually, "How do I get started?" Unless the system is quite similar to other previously used software applications, a new user seldom has either knowledge of what a software system is supposed to be able to do, or the commands and procedures for making the purported capabilities of the system actually execute.

5.2.3 Getting Started - the Model Wizard

In the author's case, it was communicated verbally that a "Model Wizard" existed to begin the process of implementing a Brahms model. The Wizard is invoked within Brahms Builder, but appears to be a completely independent application, with an entirely different "look and feel." It promises the user the ability to create the formal Brahms concepts of a model without any manual coding. The Wizard's primary problem is its incomplete state. The author's experience with each of the Wizard's three screens follows.

1. The Wizard's first screen states: "Create the concepts that you would like to have in your model. Choose the concept type first." But only one concept is available in the drop-down list, "Agent." Having created one or more Agents, merely by typing in names for them, the user may choose 'Next' to move to screen 2.

2. This screen states: "Create the activities for each concept you created in the previous step. Choose the concept type first. This will display the available concepts. Select a concept and create activities for that concept by entering a name, duration and priority for each activity and adding it to the activity list. Then create concepts of that type by entering a name and adding it to the concept list." At this early stage, the author had no idea how an "Activity" related to any other concept, such as the Agent(s) just created. The strategy chosen to move ahead was to create one agent, since that was the only concept available and also readily understood, and then to create some actions using the Create Activity screen that she perceived as interesting for the given scenario. Having done this, moving ahead to the last screen was possible.

3. The last screen states: "Give a name for your model that you are about to generate. Also give the filename and the location for the file in which to store the model." A couple of attempts clarified that this means that "the model" may have a different name from "the file", and that the filename should be typed in WITHOUT the ".b" extension, which Builder will add. A sample of code generated by the Wizard is shown in Figure 5.1

Confusingly, the resulting file from the Wizard's hidden work, chosen_name.b, contains far more than code for the Brahms agents created within the Wizard, as can be seen in the figure. The Wizard added everything after the following line:

```
} // agent FirstAgent
```

It is not obvious why the formal concepts and classes added automatically by the Wizard are essential to every Brahms model, which would be a reasonable assumption since they *are* added automatically. The beginning Brahms user, particularly anyone who is not already thoroughly familiar with C or C++ programming, may not understand why these additional constructs were added to her/his first model, or what to do with them.

```
agent FirstAgent {
    activities:
        primitive_activity AnyActivity() {
            priority: 50;
            max_duration: 100;
        } // primitive_activity AnyActivity
    workframes:
        workframe wf_AnyActivity {
            do {
                AnyActivity();
            } // do
        } // workframe wf_AnyActivity
} // agent FirstAgent

agent ANY-AGENT {
} // agent ANY-AGENT

class CClock {
    attributes:
        public int hour;
        public int minute;
        public int second;
} // class Cclock

class CDate {
    attributes:
        public int dayOfMonth;
        public int month;
        public int dayOfYear;
} // class CDate

object Clock instanceof CClock {
} // object Clock

object Date instanceof CDate {
} // object Date

areadef Building {
} // areadef Building

areadef City {
} // areadef City

areadef World {
} // areadef World
```

Figure 5.1 – Model Wizard output

Also quite confusingly, Maarten Sierhuis recommends and follows the practice himself that a Brahms modeler should separate Agents, Objects, geography definitions and so on into different "something.b" files. In later stages of the author's process of attempting to model the 15-Minute story in Chapter 4, use of the Model Wizard was abandoned altogether, as Sierhuis' practice of separating concepts is definitely advantageous for anything beyond a trivial model.

5.2.4 Next Stage - Brahms Builder

Brahms Builder seems to be rather more "polished" than the Wizard although it is difficult to precisely define why. The two interact well in that as soon as the Wizard has completed its task, the user is presented with the model just created, in Builder. Figure 5.2 shows a screen shot of Builder after Wizard created the code in Figure 5.1, with many of the clickable menu items opened up before their picture was taken.

The reader may well ask, after looking at Figure 5.2, "What goes in that large gray area on the right side?" The short, unkind answer is, "Nothing." A far more charitable explanation is, "That area is reserved for future use." The Brahms team plans to develop Builder into a full-featured programming environment, the windows and menus are already set up for the happy day when those goals can be accomplished.

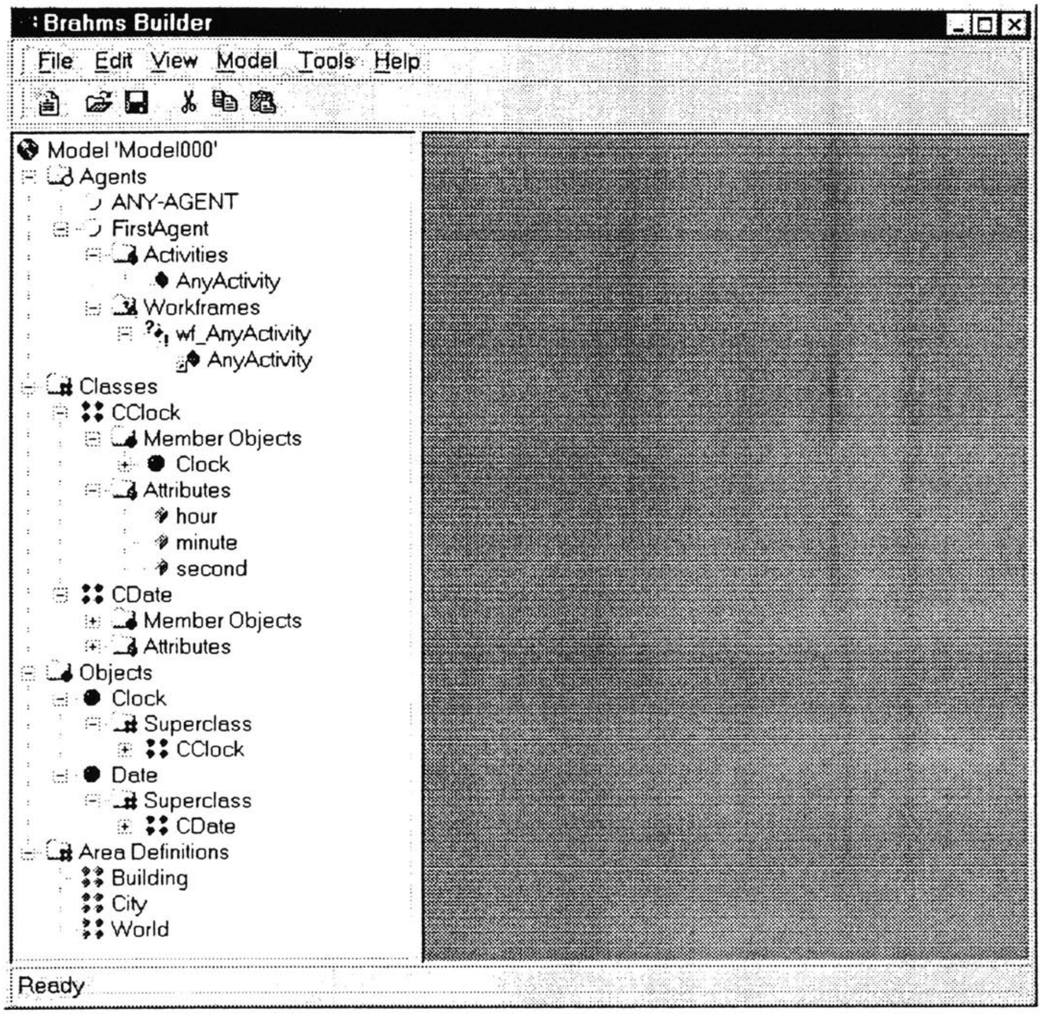


Figure 5.2 – Builder Displaying Wizard Results

After playing with Builder for a while, the user may be ready to move on.

5.2.5 The G2 Engine and its Output

The next stage in the Brahms modeling process is to invoke the model in the "Engine." The very name implies that this is the "Big Gun" of the whole system. Indeed, to get to the point of being able to perform the next step in G2, the user will have first:

- Acquired a G2 license (reportedly, approximately $15,000 at normal market rates, although Gensym Inc. President Bob Moore very kindly provided the author with a two-year license for educational research purposes at no cost.)

- Learned how to load and run *anything* in G2. Again, Gensym kindly provided the author, at no cost, enrollment in "G2-101", the full-week introductory class which begins the process of training for the use of G2.

As discussed already, within G2, Brahms appears to use few of the most obvious available modeling and simulation features that a G2 user learns about in G2-101. The Brahms engine knows how to load the "something.kb" files that Builder created, and the user may look at the parts of the engine as developed within G2. Without detailed knowledge of both G2 and Brahms, few of the visible features provide much, if any, intellectual illumination regarding the Brahms system. Upon following (undocumented) instructions to run the simulation, G2 dutifully scrolls screenfuls of text messages within one of its windows. After that, the user may issue commands to export the text file that captures the result of all this work by wo/man and machine. As noted previously, this text file is parsed to create the output database. Interestingly, G2 includes features to link its system with robust relational database systems, but Brahms, again, does not use these features.

5.2.6 Agent Viewer and the Database

Agent Viewer, the visualization portion of Brahms output, is a far more complete software application than either Wizard or Builder. Unfortunately (but not meant unkindly), it is poorly laid out, difficult to read, and rather non-intuitive. Since this is actually true for many commercially profitable applications at the moment of first use by a neophyte, this critique is perhaps actually a complement, given that Brahms is a research tool meant for use in research settings. At least it *does* have a user interface rather more developed than a blinking cursor on a command line, as many software tools in such settings typically provide.

One could quibble with a minor but confusing feature of Builder and Agent Viewer as a system, illustrated by Figures 5.3 and 5.4 (next page). Builder refers to "simulation runs" but apparently does nothing with them, while Agent Viewer does not refer to them specifically, but is the application that must process them.

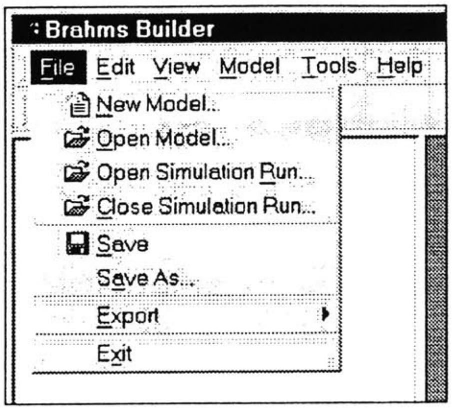


Figure 5.3 —Builder

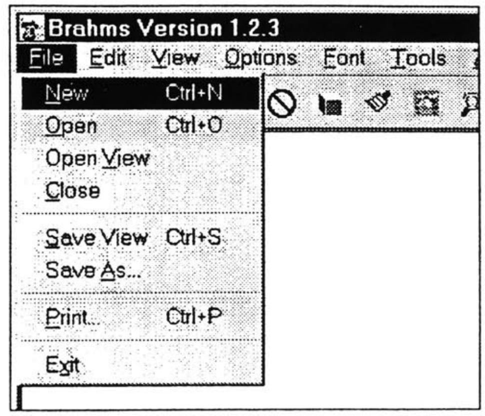


Figure 5.3 —Agent Viewer

As shown in Figure 5.3, Builder's file menu lists "Open Simulation Run" and "Close Simulation Run" items—but these are not used to process simulation runs. Agent Viewer, (the window of which is labeled "Brahms Version 1.2.3"), processes simulation runs *after the user has selected "New" in the File menu*—there is no mention of simulation runs.

After successfully processing a simulation run and permitting twiddling of its many complex layers, Agent Viewer impresses a user who has had further experience with it. It appears to make *all* of the data in the massive output database accessible and meaningfully related within the Brahms framework.

The Brahms output database, for anyone who has developed applications using Microsoft Access or another relational database system, inspires definite enthusiasm. One has no idea exactly what all those tables and queries do, but clearly there is terrific power implied by so much data, collected and presented in such carefully organized fashion.

As previously noted, since the Brahms Agent Viewer application creates output visualizations entirely from the database, it is conceivable that the Brahms system could be readily extended by using the same database for other purposes. Examples of other purposes include:

- different abstractions or visualization formats from that provided by Agent Viewer
- statistical analysis and reporting
- financial, facility management, personnel management, scheduling, operations, and many other kinds of reports
- 3D animated visualizations, e.g. IBM Data Explorer.
- stop-motion analysis viewing systems
- Virtual Reality Modeling Language (VRML) input

The database can be viewed as a springboard for many different interesting future development possibilities.

5.2.7 From Story to Code toward Output?

The author experienced each phase of the Brahms modeling process as described in the previous sections. Code forming a complete Brahms model was compiled in the Builder. Builder output was loaded into the G2 Engine, a simulation run executed, text data exported. Such data was processed with Agent Viewer, creating the database, which was reviewed, and the same data examined in Agent Viewer. These phases were experienced with code, model files or simulation runs developed by Maarten Sierhuis.

Unfortunately, the author was unable to experience the complete development process from start to finish for the MARS project of interest here. A question was formulated, the same one as in Chapter 3, section 3.14. The representative story was developed as described in Chapter 4 to inform the structure of a model that perhaps would begin to provide answers to the question.

Brahms code was created for the MARS project model, part of this code may be viewed in the Appendix. The code chosen for listing here is the "cleanest" code available after the study ended. This code was partially created with Builder, but the multi-concept files created by Builder were segmented and cleaned up to conform to Sierhuis' practice as noted previously. This strategy, in the author's opinion, *will* lead to a Brahms model of the MARS project in time. However, the complete process described here consumed much of the time available for the present study before a workable strategy to produce executable Brahms code became apparent.

Other methods not involving the Wizard—direct code writing with a text editor—were attempted when it became clear that the Wizard could not produce a complete model—even a small, simple one—which could be executed in the Engine. These other attempts began with code from complete models developed by Sierhuis. Selected code segments were modified in attempts to transform them to represent the MARS habitation scenario and the questions of interest discussed here. Several attempts were made with complete working base models, other attempts used files and code segments culled from several different models. None of these hybrid models would compile, and thus could not be executed in the Engine. Very slight modifications to the working base code would render it unable to be compiled. This was probably due to the following factors:

1. The vast differences between the objectives of the attempted new model and the base model or code segments.
2. The author's inexperience with the Brahms system and time constraints.
3. Documentation deficiencies already noted which made it extremely difficult, if not impossible, to debug or otherwise track down errors.
4. The development team's time constraints while the author faced deadlines. Due to responsibilities as NASA employees, the development team was understandably unable to serve in the user support or helpdesk role.

5.3 Brahms Language Development Recommendations

It is recommended that the team implement in a version of the Brahms language specification, if not in a software application, the following necessary notions for this field:

- All "areas" are actually volumes, with definite (and sometimes changeable) dimensions, shapes and other attributes that are essential information for understanding human interactions within. The attributes of areas include non-physical constructs such as intentions or restrictions of use, safety, comfort, adjacencies, connections, egress, maintainability and ownership. The last item, ownership, is of particular interest in environments where stress could be a major behavioral variable. This is because ownership of space is clearly associated with an individual's sense of personal control, which mediates the stress response. It is worth noting here that "sick" buildings have quite often provided no physical evidence that explains why their occupants perceive them in this way. In some cases, stress resulting from a social environment is attributed to stress caused by the enclosing physical environment.

- All "areas" are actually also objects, in that physical materials of some sort must form any enclosure. The more complex, adaptive, "intelligent" an enclosed area (or the collection of enclosed areas called a building), is, the more crucial is information about objects, controls, and systems installed beneath, above or within the surfaces that occupants ordinarily perceive as "rooms." Note also that some areas in buildings (or ships) may be critical for the functions of other areas, but ordinarily inaccessible to "routine" access or habitation. These include crawl spaces, chases, equipment rooms, ductwork, etc. Clarification would be welcome about how, or if, "Areas" may contain or otherwise be associated with "Objects" or "Conceptual Objects."

- In life-support situations, the very air is an object, but one which permeates and/or flows through all areas and many other objects. In most built-environment cases, it does so partly controlled and partly uncontrolled. On Earth, "outside" air is usually scarcely different from "inside" air, but on Mars, the boundary between these is vital to investigate and understand fully. Some discussion and guidance about this subtle but important issue would be appreciated.

- "Paths" are also actually objects, with far more necessary-to-understand attributes than origin, destination and distance. Wayfinding, for example, which may be loosely defined as the process of developing the "conceptual maps" that human beings use to find their way as they move about, demands far more information about paths than Brahms can currently accept or provide. Doors ladders, hatches and stairs, for another example, are objects that are also always intimately part of some path.

5.4 Brahms Software Development Process Recommendations

The Brahms software development team, to put it simply, really needs a *team*, and a *process*. The best programmer in the world, (which the author agrees with Clancey that Ron van Hoof probably is), simply can not develop every line of code for the robust development and application environment that the Brahms language deserves.

Funding would help, of course. The clear, if unpalatable, choices for the Brahms team are to either obtain funding for an appropriate software engineering project—complete with Requirements Analysis and Specification, Design, multiple programmers doing the coding, Configuration Management, Testing, Validation and Verification, and Quality Assurance, the "whole nine yards"—to implement the language, or to stop altogether talking and writing about Brahms as a *software* system. The present collection of code makes an excellent prototype or proof-of-concept for such a project.

References

Amato, Ivan. 1994. The sensual city. *New Scientist*, 1947 (October): 32-36.

Anonymous. 1992. Intelligent buildings in Japan. *Industries in Transition, source Focus Japan.* (December) Electronic document © Business Communications Company, Inc.

Beck, Paul E. 1993 (a). Intelligent design passes IQ test. *Consulting Specifying Engineer*, vol. 13, no. 1 (January), p. 34-38.

Beck, Paul E. 1993 (b). Measuring productivity: The Rennselaer study. *Consulting Specifying Engineer*, vol. 13, no. 1 (January), p. 35.

Beheshti, M.R. & Cser, J.A. 1992. Facilitating the management of complex buildings. *Delft Progress Report*, vol. 15, no. 2 (April), p. 65-72.

Bell, C. Gordon. 1992. Computer expert and author of *High-Tech Ventures*, quoted in March-April issue of *The Futurist*, p. 51.

Bers, Joanna Smith. 1994. Fiber optics & access floors draw Brooklyn Union to MetroTech. *Facility Design & Management*, vol. 13, no. 1 (January), p. 38-43.

Brown, Joseph S. 1991. Interactive graphics and facilities management. In *Proceedings from Facilities '91: Computer-Aided Facility Management and High-Tech Systems Conference*, Washington D.C., p. 1-5.

Brown, Malcolm. 1992. Office environment 2: Buildings and brains. *Management Today* (February), p. 82-86.

Caffrey, Ronald J. 1988. Intelligent Buildings : Fact or Fantasy. *Consulting Specifying Engineer*, vol. 4, no. 3 (September), p. 40-44.

Clancey, William J. 1991. Why today's computers don't learn the way people do. In: P.A. Flach & R.A. Meersman (eds.) *Annual Meeting of the American Education Research Association*, April. 53-62. Boston. Amsterdam.

Clancey, William J. 1993. The knowledge level reinterpreted: Modeling socio-technical systems. In: K.M. Ford & J.M. Bradshaw (eds.) *Knowledge Acquisition as Modeling*. John Wiley & Sons, New York: 33-50.

Clancey, William J. 1994. Situated cognition: How representations are created and given meaning. In: R. Lewis & P. Mendelsohn (eds.) *Lessons from Learning*. Amsterdam, North Holland: 231-242.

Clancey, William J. 1995. AI: Inventing a new kind of machine. *Computing Surveys* 27 (3) 320-322.

Clancey, William J. 1995. Practice cannot be reduced to theory: Knowledge, representations, and change in the workplace. In: S. Bagnara, Zuccermaglio C. & S. Stucky (eds.) *Organizational Learning and Technological Change*. Springer, Berlin: 16-46.

Clancey, William J. 1997. The conceptual nature of knowledge, situations, and activity. In: P. Feltovich, R. Hoffman & K. Ford (eds.) *Human and Machine Expertise in Context*. AAAI Press, CA: 247-291.

Clancey, William J. 1998. Interactive coordination processes: How the brain accomplishes what we take for granted in computer languages. In: Z. Pylyshyn (ed.) *Constraining Cognitive Theories: Issues and Options*. Greenwich: Ablex Publishing Corporation: 165-190.

Clancey, William J. 1998. Expedition To The Arctic, Haughton - Mars Project. http://www.arctic-mars.org/FIELD/july21.html

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

Clancey, William J. November 19, 1999. Human Exploration Ethnography of the Haughton-Mars Project 1998-99. http://home.att.net/~wjclancey/HCC_at_HMP-99_Mars_Society_.pdf

Clancey, William J. 1999. Visualizing practical knowledge: The Haughton-Mars Project. In: Christa Maar (ed.) *Proceedings of Munich 3rd Millennium Conference on Knowledge and Visualization.* BurdaCom, Munich, Germany.

Clancey, W. J., Jordan, B., Sachs, P. & Torok, D. 1993. Formal modeling for work systems design. *AAAI National Conference.* Washington, DC.

Clancey, W. J., Sachs, P., Sierhuis, M. & van Hoof, R. 1998. *Brahms: Simulating Work Practice.*

Clapp, M.D. & Blackmun, G. 1992. Automatic engineering of building management systems. *GEC Review*, vol. 8, no. 1 (January), p. 40-6.

Coen, Michael H. (ed.). 1998. *Proceedings of the 1998 AAAI Spring Symposium on Intelligent Environments.* AAAI Technical Report SS-98-02. Stanford, California.

Coen, Michael H. 1998. Design Principles for Intelligent Environments. *Proceedings of the Fifteenth National Conference on Artificial Intelligence.* AAAI-98. Madison, Wisconsin.

Coen, Michael H. 1999. The Future Of Human-Computer Interaction or How I learned to stop worrying and love My Intelligent Room. *IEEE Intelligent Systems.* March/April.

Cohen, Sheldon, Gary W. Evans, Daniel Stokols, and David Krantz. 1986. Chapter 3, Personal control and environmental stress. *Behavior, Health and Environmental Stress.* New York: Plenum Press.

Cross, Thomas B. 1985. Intelligent buildings: The business case. *Journal of Property Management*, vol. 50, no. 2 (March), p. 6-7,9.

Cross, Thomas B. 1986. Expert systems and the intelligent building. *Journal of Property Management*, vol. 51, no. 4 (July), p. 70-72.

Cross, Thomas B. & Blackman, Craig R. 1984. Intelligent buildings come of age. *Telecommunication Products & Technology*, vol. 2, no. 12 (December), p. 1,22-24.

Court, M.C. 1999. Commercial Aircraft-Cabin Egress: The Current State of Simulation Model Development and the Need for Future Research. *Simulation, Journal of the Society for Computer Simulation,* vol. 73, no. 4, pp 218-231.

Cox, Nick. 1986. 'Intelligent' building all wired up for a bright future. *Asian Business*, vol. 22, no. 8 (August), p. 32.

CSR writer (anonymous). 1993. Munich airport wins Intelligence award. *Consulting Specifying Engineer*, vol. 14, no. 7. (December).

DeMarco, Anthony. 1993. OAI's high-tech facility fosters collaborative research. *Facilities Design & Management*, vol. 12, no. 11 (November), p. 34-39.

Fink, Pamela K. 1992. An intelligent facility planning advisor. *Artificial Intelligence for Engineering, Design, and Manufacturing, ISA Transactions*, vol. 31, no. 2, p. 77-95.

Finley, Marion R. Jr. & Akimura, Haruo. 1991. Intelligent buildings: Myth, reality, or wishful thinking? *IEEE Communications Magazine*, vol. 29, no. 4 (April), p. 10-12.

Foley, James D., Andries van Dam, Steven K. Feiner and John F. Hughes. 1990. *Computer Graphics, Principles and Practices, Second Edition*. New York: Addison-Wesley.

Forde, Kevin. 1985. Big brother buildings. *Modern Office*, vol. 24, no. 4 (May), p. 14-15.

Gannes, Stuart. 1984. The bucks in brainy buildings. *Fortune* (December), p. 132-144.

Gallo, Fernanda. 1989. Italy explores building integration. *Consulting Specifying Engineer*, vol. 6, no. 1, Supl. (July)

Gassert, Sig. 1990. Intelligent buildings and beyond. *Telecommunication Journal of Australia*, vol. 40, no. 3 (March), p. 3-9.

Geissler, Richard. 1989. The evolution of building intelligence. *Consulting Specifying Engineer*, vol. 6, no. 1 Supl., (July) p. 4-6.

Ulyas, Laszlo, Tamas Kozsik and John B. Corliss. 1993. The Multi-Agent Modeling Language and the Model Design Interface. *Journal of Artificial Societies and Social Simulation*, vol. 2, no.3, http://www.soc.surrey.ac.uk/JASS/2/3/8.html.

Hamann, John R. 1985. Latest wrinkle for the office: Tenant services. *Office*, vol.101, no. 3 (March), p. 110-114.

Hedge, Alan. 1991. Design innovations in office environments. In *Design Intervention: Toward a More Human Architecture*, ed. Wolfgang Preiser, Jacqueline Vischer and Edward White, p. 301-321. New York: Van Nostrand Reinhold.

Hedge, Alan, William Erickson and Gail Rubin. 1995. Psychosocial correlates of sick building syndrome. *Indoor Air*, vol. 5, p. 10-21.

Hedge, Alan, William Erickson and Gail Rubin. 1993. Why do gender, job stress, job satisfaction, perceived indoor air quality and VDT use influence reports of the sick building syndrome in offices? *In Work With Display Units '92, Selected Proceedings of the 3rd International Conference WWDU '92, Berlin*, p. 49-53.

Hedge, Alan. 1990 a. Sick building syndrome correlates with complex array of factors. *Facility Management Journal*, (Jan/Feb), p. 52-58.

Hedge, Alan. 1990 b. Questionnaire design guidelines for investigations of "sick" buildings. In *Proceedings of Indoor Air '90, the 5th International Conference on Indoor Air Quality and Climate*, Toronto, Canada, (July 29-August 3), vol. 1, p. 606-610.

Hedge, Alan and Dana Ellis. 1991. New graphic database software for managing facilities performance. In *Proceedings from Facilities '91: Computer-Aided Facility Management and High-Tech Systems Conference*, Washington D.C., p. 64-77.

Hedge, Alan and Dana Ellis. 1990. Computer-aided facilities diagnostics: a new software tool for the armory of the macro-ergonomist. *Proceedings of the Human Factors Society 34th Annual Meeting*, Orlando, Florida, vol. 2, p. 859-863.

Heller, Robert J. 1990. A day in the life of an intelligent building. *Journal of Property Management*, vol. 55, no. 4 (July), p. 53-55.

Herbst, Kris. 1989. Intelligent buildings may be a smart idea after all. *Network World*, vol. 6, no. 42 (October), p. 1,55-5, 66-67.

Hopper, Andy. 1992. The walk-and-wear office. *Computerworld*, vol. 26, no. 16 (April), p.99-101.

Horitake, Hidehiro. 1990. Intelligent buildings displaying diverse development: Looking at developments in the NTT group. *NTT Review*, vol. 2, no. 3 (May), p. 62-66.

Hubbard, Gerald. 1985. A new concern: impact on people and buildings. *Office*, vol. 101, no. 1 (January), p. 150-152.

Hunt, Sherry. 1994. Getting a handle on intelligent buildings. *Building Operation Management, Supplement: Intelligent Buildings*, vol. 40, no. 8 (August), p. I/2.

Huston, Dryver R. & Fuhr, Peter L. 1993. Intelligent materials for intelligent structures. *IEEE Communications Magazine*, vol. 31, no. 10 (October), p. 40-44.

IBI. 1994. Intellex award nominees for 1992 and 1993. Faxed document. Intelligent Building Institute.

Jacobs, David M. 1985. New shared tenant systems call for new telecom management attitude: service is supreme. *Communication Age*, vol. 2, no. 6 (June), p. 24-27.

Johnson, Julie. 1994. Beating bedlam. *New Scientist*, no. 1947 (October), p. 44-47.

Karat, Clare-Marie, Arnold Lund, Joelle Coutaz, and John Karat (eds.) 1998. *Human Factors in Computing Systems, CHI '98 Conference Proceedings, (April 18-23), Making the Impossible Possible, Los Angeles, CA.* New York: Association for Computing Machinery.

Keller, Peter R. & Keller, Mary M. 1993. *Visual Cues: Practical Data Visualization*. IEEE Computer Society Press.

Kelly, Frank & Borthwick, James A.S. 1986. Integrated building systems: A case study. *Cost Engineering*, vol. 28, no. 11 (November), p. 16-25.

Kendler, Hayden. 1991. Current and future practices in advanced building control. *Electrotechnology*, vol. 2, no. 3 (June), p. 100-3.

Kreider, Jan K. & Wubbena, William H. 5/1991. Expert systems for facility management in commercial buildings. *Proceedings from Facilities '91, Computer-Aided Facility Management & High-Tech Systems Conference*, p. 79-93.

Kroner, Walter M. 1989. The new frontier: Intelligent architecture through intelligent design. *Futures*,	vol. 21, no.4 (August), p.319-333

Kujuro, Akihiko. & Yasuda, Hiroshi. 1993. Systems evolution in intelligent buildings. *IEEE Communications Magazine*, vol. 31, no. 10 (October), p. 22-26.

Lam, Stephen. 1990. Intelligent buildings. *Hong Kong Engineer*,(October), p. 32-5.

Larson, Kent. 1999. House_N: The MIT Home of the Future. http://architecture.mit.edu/~kll/

Lee, Eleanor S. & Selkowitz, Stephen E. 1993. Integrated design yields smart envelope and lighting systems. *Consulting Specifying Engineer*, vol. 13, no. 8 (June), p. 64-70.

Lee, Pascal. 1999. The Mars Arctic Research Station (M.A.R.S.): 1999 Project Status and Site Selection at Haughton Crater, Devon Island, Nunavut, Canada. Presented at the Mars Society's Second Annual Convention, August 12-15, 1999. Moffett Field, CA: National Aeronautics and Space Administration.

Loveday, D.L. & Virk, G.S. 1992. Artificial intelligence for buildings. *Applied Energy*, vol. 41, no. 3 (March), p. 201-21.

Lumenthal. Brad. Juri Gornostaev. and Claus Unger (eds.). 1995. *Human-Computer Interaction 5th International Conference, EWHCI '95, Moscow, Russia, July 3-7, 1995 Selected Papers (Lecture Notes in Computer Science, Goos et al, ed.)*. Berlin: Springer.

Mars Society. 1999. http://www.marssociety.org/

Mehta, Pratap & Clark, Gary & Thomson, Tom. 1992. Intelligent buildings: into the future. *Electrotechnology*, vol. 3, no. 3 (June). p. 25-6.

Mitchell. Mark. 1992. Intelligent buildings: a question of standards. *Electrotechnology*, vol. 3, no. 3 (June). p. 14-16.

Mitsui. Yasuhisa. 1991. Beyond intelligent buildings to facility management. *Business Japan* (October), p. 79-88.

Mool, Dennis. 1989. Systems integration improves performance. *Consulting Specifying Engineer,* Supl., (July)p. 20-21.

Moravek. James M. & Cameron. Nancy S & Olson. Kermit L., et al. 1993. Wired for change: The office environment of the future. *Consulting Specifying Engineer*, vol. 14, no. 5 (October), p. 36-44.

NASA Computational Sciences Division. 1998. *Human-Centered Computing*. http://ic.arc.nasa.gov/ic/HCC.html.

NASA Human Exploration Operations Team. 1999. *Operations Concept Definition for the Human Exploration of Mars, First Edition*. DD-099-05 April 8, 1999. National Aeronautics and Space Administration: Ames Research Center. Lyndon B. Johnson Space Center.

Neilsen. Jakob (ed). 1995. *Advances in Human-Computer Interaction Volume 5*. Norwood, New Jersey: Ablex Publishing Corporation.

Ntuen, Celestine A. & Eui H. Park (eds.) 1996. *Human Interaction with Complex Systems: Conceptual Principles and Design Practice*. Boston: Kluwer Academic Publishers.

Oliverson. Robert L. 1991. A roundtable: Trend leaders explore the status of Intelligent buildings. *Consulting Specifying Engineer*, vol. 10, no. 3 (September), p. 60-70.

Patty, Bruce R. & Gerald Hubbard. 1985. Architects must rethink their building designs, a new concern: impact on people and buildings. *Office*, vol. 101, no. 1 (January), p. 117,150-152.

Painter, Tony. 1993. Quantum House wires the intelligent building. *Communications News*, vol. 30, no. 7 (July), p. 16-19.

Pauers. William. 1988. System Integration Means Intelligent Buildings. *Consulting Specifying Engineer*, vol. 3, no. 4 (April). p. 56-60.

Pentland. Alex (Sandy). 1996. Smart Clothes. *Scientific American*, Vol. 274, No. 4, pp. 73, April 1996.

Pentland. Alex (Sandy). 1996. Smart Rooms. *Scientific American*, Vol. 274, No. 4, pp. 68-76, April 1996.

Pentland. Alex (Sandy). 1998. Wearable Intelligence. *Scientific American*, Vol. 276, No. 1es1, Nov. 1998.

Picard. Rosalind. 1998. *Affective Computing*. Cambridge, MA: The MIT Press.

Post, Nadine. 1993. Smart buildings make good sense. *Engineering News Record*, vol. 230, no. 20 (May 17), p. 24-28.

Preiser, Wolfgang. 1991. Design innovation and the challenge of change. In *Design Intervention: Toward a More Human Architecture*. ed. Wolfgang Preiser, Jacqueline Vischer and Edward White, p. 335-351. New York: Van Nostrand Reinhold.

Preiser, Wolfgang., ed. 1989. *Building Evaluation*. New York: Plenum.

Preiser, Wolfgang., H.Z. Rabinowitz and E.T. White. 1988. *Post-Occupancy Evaluation*. New York: Van Nostrand Reinhold.

Richardson, Ken. 1991. *Understanding Intelligence*. Milton Keynes & Philadelphia: Open University Press.

Robison, Rita. 1992. Smart structures. *Civil Engineering*, (November) p. 66-68.

Rubin, Arthur. 1991. Intelligent building technology in Japan. *U.S. Department of Commerce Publication.*

Rumbaugh, James & Michael Blaha, William Premerlani, Frederick Eddy, William Lorensen. 1991. *Object-Oriented Modeling and Design*. Englewoods Cliff, New Jersey: Prentice Hall.

Sinopoli, James. 1985. New opportunities in tenant telecommunications systems. *Journal of Property Management*, vol. 50, no. 2 (March/April), p. 22-25.

Senders, Cherri. 1992. Shock steady: smart buildings guard against bad vibrations. *Omni*, vol. 14, no. 8 (May), p. 26.

Serro, C. & Nunes, R. 1991. Services architecture in SEIS - A system for intelligent buildings. *Proceedings of the 6th Mediterranean Electrotechnical Conference - Melecon '91*. Piscataway, NJ: IEEE, p. 558-561.

Selvin, A.M., M. Sierhuis. (1999). Argumentation in Different CSCA Project Types, *Workshop on Computer-Supported Collaborative Argumentation for Learning Communities, Computer Supported Collaborative learning (CSCL) Conference '99*, Stanford University, CA

Sierhuis, M & A.M. Selvin (1996). Towards a Framework for Collaborative Modeling and Simulation. Presented at the *Workshop on Strategies for Collaborative Modeling and Simulation*, CSCW'96 conference, Boston, MA.

Sierhuis, M. & Clancey, W. J. (1997) Knowledge, Practice, Activities and People.

Sierhuis, M. (1996). Selective Ethnographic Analysis; Qualitative Modeling for Work Place Ethnography. Presented during the *American Association of Anthropology Conference 1996*, San Francisco, CA.

Sierhuis, M. and W. J. Clancey (1997). Knowledge, Practice, Activities, and People. *Proceedings of AAAI Spring Symposium on Artificial Intelligence in Knowledge Management, Stanford University*, CA.. http://ksi.cpsc.ucalgary.ca/AIKM97/AIKM97Proc.html

Sierhuis, M., W.J. Clancey, R. van Hoof. (1999). BRAHMS: A multi-agent programming language for simulating work practice. RIACS/NASA Ames Research Center

Simmonds, R. 1992. Smart buildings rise in the East. *Electrical Review*, vol. 225, no. 14 (July), p. 74-75,77,79.

Sracel, Holly. 1993. Can flexibility and wire management coexist? *Facilities Design & Management*, vol. 12, no. 4 (April), p. 44-47.

Sracel, Holly. 1993. Salomon's White links FM, technology & teamwork at WTC HQ. *Facilities Design & Management*, vol. 12, no. 9 (September), p. 56-59.

Sullivan, Joseph W., Sherman W. Tyler (eds.). 1991. *Intelligent User Interfaces*. New York: ACM Press.

Takahashi, Kimikaza, & Tom Kateeshock. 1992. Expert system for refinery off-site facility management.	 *Artificial Intelligence for Engineering, Design, and Manufacturing, ISA Transactions*, vol. 31, no. 2 (March), p. 77-75.

Tateishi, Makoto. 1988. Making the best use of intelligent buildings. *Business Japan*, vol. 33, no. 10 (October), p. 41-51.

Tateishi, Makoto. 1989. Intelligent buildings must meet needs of tenants. *Business Japan*, vol. 34, no. 10 (October), p. 55-62.

Tufte, Edward R.. 1983. *The Visual Display of Quantitative Information*. Cheshire, Connecticut: Graphics Press.

Tufte, Edward R.. 1990. *Envisioning Information*. Cheshire, Connecticut: Graphics Press.

Umeno, Shoichiro. 1993. The continuing development of facility management in Japan. *Japan 21st*, vol. 38, no. 10 (October), p. 41-49.

Uras, H M. & Aktan, H M. 1993. A hybrid energy-dissipation device for intelligent buildings. *Experimental Mechanics*, vol. 33, no. 1 (March), p. 15-20.

van Hoof, Ron, William Clancey & Maarten Sierhuis. November 17, 1999. *BRAHMS Language Specification Version 1.8 – Final: Technical Memorandum TM99-0008*. Moffett Field CA: National Aeronautics and Space Administration.

Watt, Alan. 1993. *3D Computer Graphics, Second Edition*. New York: Addison-Wesley.

Webb, R. 1990. Management applications of computers in intelligent buildings. *Proceedings of the Winter Meeting, American Society of Mechanical Engineers.*

Weingarten, Nicholas H. & Evans, Paul Mark. 1991. Artificial intelligence for facility management. *Proceedings from Facilities '91, Computer-Aided Facility Management & High-Tech Systems Conference*, p. 111-125.

Wickens, Christopher, David H. Merwin and Emile L. Lin. 1994. Implications of graphics enhancement for the visualization of scientific data: dimensional integrality, stereopsis, motion, and mesh. *Human Factors*, vol. 36, no. 1, p. 44-61.

Wolff, Rob and Larry Yaeger. 1993. *Visualization of Natural Phenomena*. New York: TELOS/Springer-Verlag.

This page intentionally left blank.

Appendix – MARS Project Brahms Model Code


```
//************************************************************
//************************************************************
//
//Mars Arctic Research Station Brahms Model Code Segments
//(c)2000 Simonetta A. Rodriguez
//
//
//begin segments
//************************************************************
//************************************************************
//Mars Arctic Research Station Brahms Model Component
//

//MarsGeography.b ------------------------------ areas and paths

// only World, City, Building available now, no areadefs possible

import brahms.base.World;
import brahms.base.City;
import brahms.base.Building;

area MarsGeography instanceof World { }

//---------------------------------------------------- on Earth
area Earth instanceof City partof MarsGeography {
   display: "Earth";
   attributes:
   relations:
   initial-facts:
}

area MissionControl instanceof Building partof Earth {
   display: "Mission Control";
   attributes:
   relations:
   initial-facts:
}

//------------------------------------------- on the planet Mars
area Mars instanceof City partof MarsGeography {
   display: "Mars";
   attributes:
   relations:
   initial-facts:
}
```

```
//---------- the landing site and permanent location of this Hab
area HabSite instanceof Building partof Mars {
   display: "HabSite";
   attributes:
   relations:
   initial-facts:
}

area RemoteArea instanceof Building partof Mars {
   display: "Remote Area";
   attributes:
   relations:
   initial-facts:
}

//----- the habitation unit (main module only, no extensions yet)

area MarsHab instanceof City partof MarsGeo {
   display: "MarsHab";
   attributes:
   relations:
   initial-facts:
}

area GroundLevel instanceof Building partof MarsHab {
   display: "DnLevel";
   attributes:
   relations:
   initial-facts:
}

area UpperLevel instanceof Building partof MarsHab {
   display: "UpLevel";
   attributes:
   relations:
   initial-facts:
}

area Entry instanceof Building partof MarsHab {
   display: "Entry";
   attributes:
   relations:
   initial-facts:
}

area GroundLevelRoom1 instanceof Building partof MarsHab
   display: "DnRm1";
   attributes:
   relations:
   initial-facts:
}
```

```
area GroundLevelRoom2 instanceof Building partof MarsHab {
   display: "DnRm2";
   attributes:
   relations:
   initial-facts:
}

area GroundLevelStoreRoom1 instanceof Building partof MarsHab {
   display:  "DnStore1";
   attributes:
   relations:
   initial-facts:
}

area GroundLevelStoreRoom2 instanceof Building partof MarsHab {
   display: "DnStore1";
   attributes:
   relations:
   initial-facts:
}

area HatchA instanceof Building partof MarsHab {
   display: "Hatch A";
   attributes:
   relations:
   initial-facts:
}

area HatchB instanceof Building partof MarsHab {
   display: "Hatch B";
   attributes:
   relations:
   initial-facts:
}

area HatchAreaA instanceof Building partof MarsHab {
   display: "A_area";
   attributes:
   relations:
   initial-facts:
}

area HatchAreaB instanceof Building partof MarsHab {
   display: "B_area";
   attributes:
   relations:
   initial-facts:
}

area UpperLevelRoom1 instanceof Building partof MarsHab {
   display: "UpRm1";
```

```
    attributes:
    relations:
    initial-facts:
}

area UpperLevelRoom2 instanceof Building partof MarsHab {
    display: "UpRm2";
    attributes:
    relations:
    initial-facts:
}

area UpperLevelRoom3 instanceof Building partof MarsHab {
    display: "UpRm3";
    attributes:
    relations:
    initial-facts:
}

area Bunk1 instanceof Building partof MarsHab {
    display: "Bunk1":
    attributes:
    relations:
    initial-facts:
}

area Bunk2 instanceof Building partof MarsHab {
    display: "Bunk2":
    attributes:
    relations:
    initial-facts:
}

area Bunk3 instanceof Building partof MarsHab {
    display: "Bunk3";
    attributes:
    relations:
    initial-facts:
}

area Bunk4 instanceof Building partof MarsHab {
    display: "Bunk4";
    attributes:
    relations:
    initial-facts:
}

area Bathroom instanceof Building partof MarsHab {
    display: "Bathroom";
    attributes:
    relations:
    initial-facts:
```

```
}

area Kitchen instanceof Building partof MarsHab {
   display: "Kitchen";
   attributes:
   relations:
   initial-facts:
}
area ComdArea instanceof Building partof MarsHab {
   display: "Command Area";
   attributes:
   relations:
   initial-facts:
}


//----------------------------------------------- Path definitions
//------------------- Path from Earth to Mars
path toMars {
   display: "toMars";
   area1: Earth;
   area2: Mars;
   distance: 9999999999999999;
}

//------------------- Path from Mars to Earth
path toEarth {
   display: "toEarth";
   area1: Mars;
   area2: Earth;
   distance: 9999999999999999;
}

//------------------- Any Path on Mars outside Hab
path Traverse {
   display: "display";
   area1: HabArea;
   area2: RemoteArea;
   distance: 600;
}

//------------------- Paths inside Hab
path Hab_Up {
   display: "Hab_Up";
   area1: GroundLevel;
   area2: UpperLevel;
   distance: 5;
}
path Hab_Dn {
   display: "Hab_Dn";
   area1: UpperLevel;
   area2: GroundLevel;
```

```
   distance: 5;
}

//*********************************************************
//*********************************************************
//Mars Arctic Research Station Brahms Model Component
//(c)2000 Simonetta A. Rodriguez
//

//Groups.b --------------------- agent functional groupings

import brahms.*;

//----------------------------------------------- Crew
group Crew memberof BaseGroup {
   attributes:
      public string Name;

   initial_beliefs:
      (the groupMembership of current = "Crew");
      (the agentLocation of current = "MarsHab");

   initial_facts:
      (current contains IDchip);

} // group Crew


group Pilot memberof Crew {
   initial_beliefs:
      (the functionMembership of current = "Pilot");
} // group Pilot

group Commander memberof Crew {
   initial_beliefs:
      (the functionMembership of current = "Commander");
} // group Commander

group Mechanic memberof Crew {
   initial_beliefs:
      (the functionMembership of current = "Mechanic");
} // group Mechanic

group Explorer memberof Crew {
   initial_beliefs:
      (the functionMembership of current = "Explorer");
} // group Explorer

group Scientist memberof Crew {
   initial_beliefs:
      (the functionMembership of current = "Scientist");
} // group Scientist
```

```
group Geologist memberof Scientist {
    initial_beliefs:
        (the functionScientific of current = "Geologist");
} // group Geologist

group Biogeochemist memberof Scientist {
    initial_beliefs:
        (the functionScientific of current = "Biogeochemist");
} // group Biogeochemist

//---------------------------------------------- Mission Control
group MissionControl memberof BaseGroup {
    initial_beliefs:
        (the groupMembership of current = "MissionControl");
} // group MissionControl

//*************************************************************
//*************************************************************
//Mars Arctic Research Station Brahms Model Component
//(c)2000 Simonetta A. Rodriguez
//

//Agents.b -------------------------------------- Crew Agents.

import brahms.*;
import Mars.Groups;
import Mars.MarsGeography;

//---------------------------- agent Bi ("Bee")

agent Bi {memberof Biogeochemist {
    display: "Bi";
    location: MarsHab;

    initial_beliefs:
        //names of self and other agents
        (the name of current = Bi);
        (the name of Eug = Eug);
        (the name of Geo = Geo);
        (the name of Skip = Skip);

    initial_facts:
        (current contains xClothing);
        (the CrewID of current = "Bi");
        (the Gender of current = "Male");
        (the PrimaryExpertise of current = "Biogeochemistry");
} // agent Bi


//Eug.b --------------------- agent Eug ("Yooj")
```

```
agent Eug {memberof Pilot, Mechanic {
   display: "Eug";
   location: MarsHab;

   initial_beliefs:
      //names of self and other agents
      (the name of current = );
      (the name of Bi = Bi);
      (the name of Eug = Eug);
      (the name of Geo = Geo);
      (the name of Skip = Skip);

   initial_facts:
      (current contains EugClothing);
      (the CrewID of current = "Eug");
      (the Gender of current = "Male");
      (the PrimaryExpertise of current = "Pilot");
} // agent Eug


//Geo.b --------------------- agent Geo ("Joe")

agent Geo {memberof Geologist {
   display: "Geo";
   location: MarsHab;

   initial_beliefs:
      //names of self and other agents
      (the name of current = Geo);
      (the name of Bi = Bi);
      (the name of Eug = Eug);
      (the name of Skip = Skip);

   initial_facts:
      (current contains GeoClothing);
      (the CrewID of current = "Geo");
      (the Gender of current = "Female");
      (the PrimaryExpertise of current = "Geology");
} // agent Geo


//Skip.b --------------------- agent Skip

agent Skip {memberof Mechanic, Commander {
   display: "Skip";
   location: MarsHab;

   initial_beliefs:
      //names of self and other agents
      (the name of current = Skip);
      (the name of Bi = Bi);
      (the name of Eug = Eug);
```

```
        (the name of Geo = Geo);

    initial_facts:
        (current contains SkipClothing);
        (the CrewID of current = "Skip");
        (the Gender of current = "Female");
        (the PrimaryExpertise of current = "Mechanics");
} // agent Skip


/* Question: what will result from Skip's and Eug's dual group
membership?  Note that the two initial groups for each carry the
same initial_belief attribute "functionMembership", but with
different values.  ToDo: model scenario to test the ambiguity.
*/

//Skip.b -------------------------------- agents at Mission Control.

agent CommOfficer memberof MissionControl {
    display: "Earth";
    location: Earth;
    initial_beliefs:
        (the functionMembership of current = "CommOfficer");
} // agent CommOfficer

agent Doctor memberof MissionControl {
    display: "Doc";
    location: Earth;
    initial_beliefs:
        (the functionMembership of current = "Doctor");
} // agent Doctor

agent Dentist memberof MissionControl {
    display: "Tooth";
    location: Earth;
    initial_beliefs:
        (the functionMembership of current = "Dentist");
} // agent Dentist


//************************************************************
//************************************************************
//Mars Arctic Research Station Brahms Model Component
//(c)2000 Simonetta A. Rodriguez
//

//Objects.b -------------------------------- Object definitions.

import brahms.base.*;

object HabObject instanceof BaseClass {
    display: "Hab Object"
```

```
   location: MarsHab;

   attributes:
      public symbol ground_elevation;
      public symbol diameter;
      public symbol height;

   // does this permanently affiliate the object with the area?
   relations:
      HabObject MarsHab;

   initial_facts:
}

object RoomObject instanceof BaseClass {
   display: "Room Object"
   location: MarsHab;

   relations:

   attributes:
      public symbol area;
      public symbol major_dim;
      public symbol minor_dim;
      public symbol height;

   initial_facts:
}
```

```
/* List of more objects and variations to define:

IDchip
BunkObject - bed within the Bunk area
Clock - alarm, voice
ControlDevice - airtemp, airvolume, airhumidity, lighting, etc.
CommUnit - varieties?
Hatch
ExtDoor
Floor
Ceiling
Ladder
Ramp
Furniture - (somewhat moveable) chair, table
Fitting - (fixed)
Lamp
Rover
Clothing
Tools
Equipment - varieties
Packages - standard groups of objects

*/

//**************************************************************
//**************************************************************
//end segments
```

This page intentionally left blank.