BRAHMS: SIMULATING PRACTICE FOR WORK SYSTEMS DESIGN

William J. Clancey
NASA/Ames Research Center, MS 269-1
Moffett Field, CA 94035 USA
Bclancey@mail.arc.nasa.gov

Patricia Sachs
Institute for Research on Learning
66 Willow Place, Menlo Park, CA 94028 USA
patricia_sachs@irl.org

Maarten Sierhuis
NASA/Ames Research Center
msierhuis@mail.arc.nasa.gov

Ron van Hoof
Bell Atlantic, Science and Technology, Inc.
400 Westchester Avenue, White Plains, NY 10604 USA
rvhoof@basit.com

ABSTRACT

A continuing problem in business today is the design of human-computer systems that respect how work actually gets done. The overarching context of work consists of *activities*, which people conceive as ways of organizing their daily life and especially their interactions with each other. Activities include reading mail, going to workshops, meeting with colleagues over lunch, answering phone calls, and so on.

Brahms is a multiagent simulation tool for modeling the activities of groups in different locations and the physical environment consisting of objects and documents, including especially computer systems. A Brahms model of work practice reveals *circumstantial, interactional influences* on how work actually gets done, especially how people involve each other in their work. In particular, a model of practice reveals how people accomplish a collaboration through multiple and alternative means of communication, such as meetings, computer tools, and written documents. Choices of what and how to communicate are dependent upon *social beliefs and behaviors*—what people know about each other's activities, intentions, and capabilities and their understanding of the norms of the group. As a result, Brahms models can help human-computer system designers to understand *how tasks and information actually flow* between people and machines, what work is required to synchronize individual contributions, and how tools hinder or help this process. In particular, workflow diagrams generated by Brahms are *the emergent product of local interactions between agents and representational artifacts*, not pre-ordained, end-to-end paths built in by a modeler.

We developed Brahms as a tool to support the design of work by illuminating how formal flow descriptions relate to the social systems of work; we accomplish this by incorporating multiple views—relating people, information, systems, and geography—in one tool. Applications of Brahms could also include system requirements analysis, instruction, implementing software agents, and a workbench for relating cognitive and social theories of human behavior.

OVERVIEW OF OBJECTIVE, THEORETICAL STANCE, AND CONTRIBUTION

Brahms is a multi-agent simulation framework (Tokoro, 1996) for modeling work practice, incorporating state-of-the-art methods from artificial intelligence research and insights about work and learning from the social sciences. A Brahms model is a kind of theatrical play, intended to provoke conversation and stimulate insights in groups of people seeking to analyze or redesign their work. Rather than modeling technical knowledge in detail, Brahms models focus on the conventions by which people choose to use particular tools and interact with each other, such as how they communicate. The quality, methods, and evaluation criteria of technical problem solving—the focus of most computer systems design—are constrained by this social-interactional context (Sachs, 1995; Schon, 1983; Weickert, 1995; Zuboff, 1987). We hypothesize that multiple, complementary views—cognitive, social, physical—integrated into one model provide a better basis for understanding organizations than cognitive task models, which are disembodied and oriented around individuals, or business process models, which are overly abstract, and hence decontextualized. More generally, we are interested in how organizations change themselves, and thus how to design a workplace so that people will dynamically reconfigure their processes, use of tools, and collaboration to creatively affect how a job gets done (Nonaka and Takeuchi, 1995). In Brahms we apply and extend knowledge-based techniques in a way that seeks to understand how information and workflow actually happens. Our approach demonstrates how symbolic cognitive modeling, traditional business process modeling, and situated cognition theories may be brought together in a coherent approach to the design of human-computer systems.

This introduction provides an overview of our objectives, theoretical stance, and contribution to human-computer system design. Subsequent sections in this paper describe the relation of Brahms to situated cognition and workflow modeling, the methodology, provide examples, present results, and analyze broader implications.

Practice, work systems design, and modeling work

Broadly speaking, *work practice* may be contrasted with *work process*; practice concerns *how people actually behave* within a physical and social environment, as distinct from the functions they accomplish. For example, a description of work practice might include who picks up a fax, where it is delivered (to a desk? to a group of mailboxes?), and when this is done. In contrast, a typical description of work process would only show that an order, for example, is sent from one organization to an agent who processes it. The faxing process and how it is carried out might not be mentioned at all. In short, a model of practice is oriented around *agents*—how they interact with their environment and what they do during the course of a day. A model of work process is typically oriented around *work products* (such as orders)—how they are transformed and flow

from one transformation to the next. A key finding of our work is that a representation of work process, such as *work flow diagrams*, can be derived from the result of a simulation of practice.

The notion of "practice" is central to work systems design, which has its roots in the design of socio-technical systems, a method developed in the 1950s by Eric Trist and Fred Emery (1959, 1960). *Socio-technical systems design* sought to analyze the relationship of the social system and the technical system, such as manufacturing machinery, and then design a "socio-technical system" that leveraged the advantages of each. *Work design* (Ehn, 1989; Greenbaum and Kyng, 1991; Pasmore, 1994; Weisbord, 1987 (see Chapter 16)) extends this tradition by focusing on both the formal features of work (explicit, intentional) and the informal features of work (as it is actually carried out "in practice," analyzed with the use of ethnographic techniques).

The aim of analyzing both the formal and the informal work practices is two-fold: to understand what it takes to actually accomplish a business function in order to use those insights in design, and to ensure that new designs of work can be effectively implemented. Socio-technical and work systems design aim at producing both a productive workplace and a positive work environment, which *fosters human development* by providing dignity, meaning and community. By contrast, business process reengineering focuses on the structuring of key business process, eliminating duplication and unnecessary steps, formulating processes and procedures and using information technology extensively to improve work processing. Consequently, business process reengineering focuses more exclusively on tasks, does not take into account the informal nature of work, and does not hold as important the dignity of work or human development. In short, work design includes a focus on practice, while business process design exclusively focuses on process (Sachs, 1993).[1]

The methods of business process reengineering (BPR) and work system design differ considerably. BPR tends to be conducted by external consultants who analyze the flow of work products in terms of business functions, and not how product get from one place to another. Work design is more typically carried out by people who actually do the work (both workers and managers), and emphasizes not only what flows, but how and why work products manage to get from one place to another. For example, managers, office workers, crafts people, and researcher-facilitators collaborate to understand an existing work setting (such as new order processing for a telecommunications company) in order to develop a comprehensive design for the business organization, work process, computer tools, documents, facilities (such as seating arrangements), training, performance metrics and incentives, etc.

[1]Both approaches may also be contrasted with an economic or business strategy analysis, which invents the products and services that both business process reengineering and work system design seek to deliver.

The two approaches differ in their view of how information technology should be used. In BPR, models of technical problem-solving, for example, tend to result in business process designs in which information technology is seen as an opportunity to "do the work" (e.g., to automate as much of the work as possible). In work systems design, information technology is seen in terms of augmenting and supporting human work practices. *These outlooks have profoundly different consequences for the design of software systems.*

We emphasize that in work design the design process attempts to treat everyday work as the result of a combination of interacting conceptual and physical influences and that practices will develop over time through learning and worker invention. Because work systems are not simply "implemented," but develop and grow through the learning of communities, workers and designers (such as software engineers) must collaborate in the design process (Greenbaum and Kyng, 1991). The differences in these two approaches reflect theoretical underpinnings about the nature of knowledge and learning (what does "knowledge of work" mean when an individual is part of the work, or not part of it all at? See discussion below on Situated Cognition).

Finding ways to effectively "see" process and practice at work has been a key challenge for us. We have developed Brahms because we think it provides a step forward in visualizing, analyzing, and thinking about the multiple dimensions of work that we think informs design. By understanding the distinction between process and practice, we realize that *organizations function at many levels and in many ways simultaneously*. Both process and practice exist, but they are not the same thing. It is therefore unrealistic to assume that one could design a process and expect the practice—the actual doing of the work—to follow flawlessly. At the same time, one should not expect to design practice and assume it will produce the process that the business needs. Brahms, as a tool to build integrated models, offers a way of seeing both process and practice.

A key finding of our work is that a representation of work process, such as *work flow diagrams*, can be derived from the result of a simulation of practice. Brahms has been developed as a tool to facilitate the representation and visualization of both process and practice. While it focuses primarily on practice, the simulated model generates a work flow which can be analyzed and discussed, and which is integrated with actual practice. This kind of model can leverage an integrated understanding of work flow by making visible both practice and process, and can close the distance between business process models and realistic implementation, which takes place in practice.

Models to support an integrated socio-technical view of work design

A model of practice is especially useful for revealing informal aspects of work. Conventionally, tools for modeling work produce either detailed descriptions of reasoning, as in *cognitive task models* (Newell, 1990), or descriptions of work product flows between job functions, as in *business process models* (Tyo, 1995). These modeling techniques cannot be used to easily or directly represent informal interactions that have a direct effect on the quality of work: how collaborative troubleshooting occurs, how learning occurs on the job, how people work on multiple orders at once, when people engage each other in each other's work, how people use communication media in practice (telephone, e-mail, face-to-face conversation, etc.), and so on. Such factors are circumstantial and conventional and cannot be strictly specified in advance (or controlled by management). Incorporating circumstantial factors in a model of work leads us to consider what people are actually doing, how these practices came about, and whether or not advantageous interactions are emerging. In contrast, models of technical problem solving (as in most cognitive models and expert systems) or formal policies and organization charts ("what should be") are of limited value for communicating to workers and managers alike what informal interactions are occurring and how computer systems, workplace layout, management attitudes, etc. facilitate or hinder such interactions.

More generally, with respect to the overall objectives of human-computer system design, Brahms provides a tool for engaging workers during the software design process, which can be a key aspect of work systems design. In contrast to business reengineering's use of information technology, *work system design aims to understand how the computer system and the human system can be most productively integrated.* As a model of how work actually occurs, Brahms helps designers understand the context in which computer tools are used (Weickert, 1995). For example, a model of practice reveals how information that is entered into a computer database is first acquired by reading a faxed form, by talking to the person in the next cubicle, or by looking up instructions in a manual. Thus, we approach human-computer interaction from a comprehensive, systemic perspective that seeks to relate how people interact with each other, kinds of representations in the environment, and physical layout of materials (Greenbaum and Kyng, 1991). As we will show, this comprehensive approach provides insights for understanding the human and social factors of software engineering, both in terms of *content* for representing work processes and in terms of *methodology* for facilitating the design process. Specifically, by providing a language for representing activities, Brahms improves upon empirical studies and user models that focus only on psychological processes. That is, Brahms models show not only what reasoning occurs, but how the information being used came to be available—based on *where* the reasoning is occurring, what other tools are being used, who is participating in the problem solving

situation, and how (because of other circumstantial physical and social factors) they came to participate.

A cognitive model would of course emphasize that the knowledge of a person affects the quality of the work produced. But a task model of work does not explain *how particular people became involved in solving particular problems.* Work systems design emphasizes the crucial process by which *people reconfigure their organization and tools* to bring resources to bear on a given situation. Specifically, who is involved in a situation assessment will determine how the situation is framed (is it a craft problem? a software problem? a management problem? a problem with policy?) and what problem solving methods are applied. In this respect, modeling practice addresses the issues that are raised by the theme of "resource-bounded reasoning" in artificial intelligence research (Zilberstein, 1996) by broadening from issues of how reasoning is managed to how the social-interactional environment determines *which agents* are involved in reasoning at all about a given problem.

In summary, our approach to practice, work systems design, and model-building in Brahms brings together psychological, social, and physical environmental factors in a coherent manner. As a multi-agent simulation program, Brahms relates traditional engineering approaches to the study of people (e.g., task models) and knowledge-based approaches for representing processes qualitatively. Specifically, agents' behaviors and attention are modeled in Brahms using a rule-based, subsumption architecture (Brooks, 1991; Nakashima, Noda, and Hana, 1996). Behaviors are organized into and inherited from groups to which agents belong; groups include not only technical functions (such as "splicer"), but where people work ("1 World Trade Center people"), their temporary roles ("turf coordinator"), their background ("new hire from outside the company"), and the tools they use ("CIMAP database users"). Most importantly, we model located behaviors of people in time and view the rule-like constructs in the model as *descriptions of what people do*, not what they know per se.

Because Brahms' design is a combination of process modeling and situated cognition ideas, additional preliminary discussion is helpful to understand what we seek to include in Brahms models (hence why other tools are inadequate) and how our methods and objectives relate to the field of artificial intelligence more broadly. The following section uses an example to relate Brahms relates to situated cognition theories.

WHAT SITUATED COGNITION REVEALS ABOUT WORK PRACTICE

This section presents our view of situated cognition and how this relates to modeling work. The main idea is that a model can be more insightful and useful for work systems design if it makes fewer assumptions about how work gets done than are built into cognitive and workflow models.

This is accomplished in Brahms by detailing and allowing variability in the description of *how information and work product flows occur* in the physical-social environment.

Overview of situated cognition

Situated cognition is an approach for understanding cognition that seeks to relate social, neural, and psychological views (Clancey, 1997b). From the social perspective, situated cognition provides insights about *the content of knowledge*, namely how people conceive of what they are doing in terms of their contribution to a community of practice (Wenger, 1997) and how this affects their attention and priorities over time. From the neural perspective, situated cognition provides insights about *the physical structure of knowledge*, namely how perception, conception, and motor action are related through a self-organizing coordination process with a memory. From a psychological perspective, situated cognition provides insights about *how behavior is improvised* by resequencing and recomposing previous behaviors.

Obviously, a great deal more could be and has been said about these topics (e.g., see Brown, Collins, and Duguid, 1988; Resnick, Levine, and Teasley, 1991; Agre 1997; Clark, 1997); the point of this overview is that situated cognition is not just a claim about "the context-dependent nature of symbolic descriptions of human knowledge" (Menzies, in press). Instead, criticisms of the "symbolic" approach to building an artificial intelligence (e.g., see Steels and Brooks, 1995) are based on a host of more fundamental claims about:

a) *the co-dependency of perception and conception* (e.g., how they may organize each other at the same time, rather than being a linear flow, perceive-> reason -> act),

b) *the coordination of speaking* (verbal descriptions) *with other modes of conception* (e.g., gesture, imagery, verbal, auditory)

c) *the transactional relation of descriptive models* (any statements, maps, knowledge bases, policies, design, recipes, rules) *to future human activity* (i.e., why descriptions must be adapted to new situations),

d) *the improvisatory, structural aspect of memory* (i.e., reconstructed, self-organizing processes, not stored programs that are merely reapplied in new circumstances),

e) *the interactive, behavioral aspect of performance* (behavior is interactive means that action changes the person and the environment; action is not merely an input/output function),

f) *the social-activity aspect of conception* (i.e., the content of knowledge is not just technical, but organized by identity and the choreography of participation; intention is not concerned only with technical goals or tasks, but becoming a member of a community of

practice; and these are broader, more pervasive influences on behavior than technical problem solving viewed narrowly as reasoning).

These considerations are much more complicated than the traditional "symbolic" view of knowledge. There are many implications for understanding human learning, the use of tools, and how to design tools. For example, situated cognition suggests that issues of identity are central to understanding a person's motivation for using a tool. From the psychological side, situated cognition reveals the interpretative, conceptual work involved in dealing with a business policy. In turn, these insights better reveal the nature of "routine" work and "problematic situations" (Schon, 1983; Wynn, 1991).

In short, a situated cognition perspective suggested that we improve business process models by representing task performance within the context of social-interactional behaviors. In this way, we are drawn to consider how psychological, social, and physical factors interact to affect what is normally taken for granted by problem solving models, namely how people make observations (gaining new information), how people prioritize their tasks while juggling multiple responsibilities, and how they decide whether and by what means to communicate information. Simply put, a straight-forward knowledge-based approach to modeling an office would suggest that people are literally following their job function descriptions, and hence doing the same thing at 9 AM as at 4:45 PM, that they are either knowledge clones or in need of training, that they are never working on other people's problems, that they do not answer a phone on someone else's desk, etc.

Models can be built at different levels to describe different phenomena. But the constructs in a modeling language often reflect assumptions about what a model should include. Specifically, the original paradigm of expert systems (Hayes-Roth, Waterman, and Lenat, 1983) assumes that knowledge is exclusively technical, objective, and used for reasoning. In the original formulation, all human action is viewed as being problem solving in an unspecified environment. At issue is not so much the modeling apparatus (though indeed more is needed to model human attention and physical interaction), but the concepts used for describing human behavior. Specifically, expert systems did not represent the conceptual and physical context in which reasoning occurs, which we call *work practice*.

The analysis of work practice originated in the social sciences, particularly anthropology and the ecological approach to sociology called *situated action* (Mills, 1940). The application to work systems design is perhaps most evident in socio-technical systems approaches in the 1950s (Emery and Trist, 1960). We were specifically most influenced by the research on situated action collected in *Design at Work* (Greenbaum and Kyng, 1991). Simply put, the situated action perspective

claims that the quality of work (including the quality of software design) depends on circumstantial, physical interaction with materials and who participates:

> The designer must reverse the field, pull the background into the foreground, and begin to see how portions of behavior function as a part of the process. The workplace doesn't consist exclusively of manipulating symbols or materials that have predictable properties under specific conditions, that can be formulaically arranged to produce a certain outcome. It is based on a background of practices, in fact, practices within practices. Imagine asking somehow how he swims. Even if he could describe the movements adequately, he probably would not mention the pre-condition: You have to be in water, alive. (Wynn, 1991)

But business process models are useful because they abstract away details. What aspects of the "background of practices" should be considered? How can modeling practice be usefully scoped and focused? That is the key problem we had to address in developing Brahms. Based on our analyses of task/workflow models (e.g., see Joosten and Brinkkemper, 1996), and given our objective of incorporating informal, circumstantial factors that influence work quality, we concluded that a model of practice should primarily 1) *represent the daily activities during which tasks are carried out* and 2) *attempt to comprehensively model when and how conversations occur*. The next subsection provides an example of the level of detail business process models omit, which we hypothesized should be included if we are to understand how collaboration actually occurs.

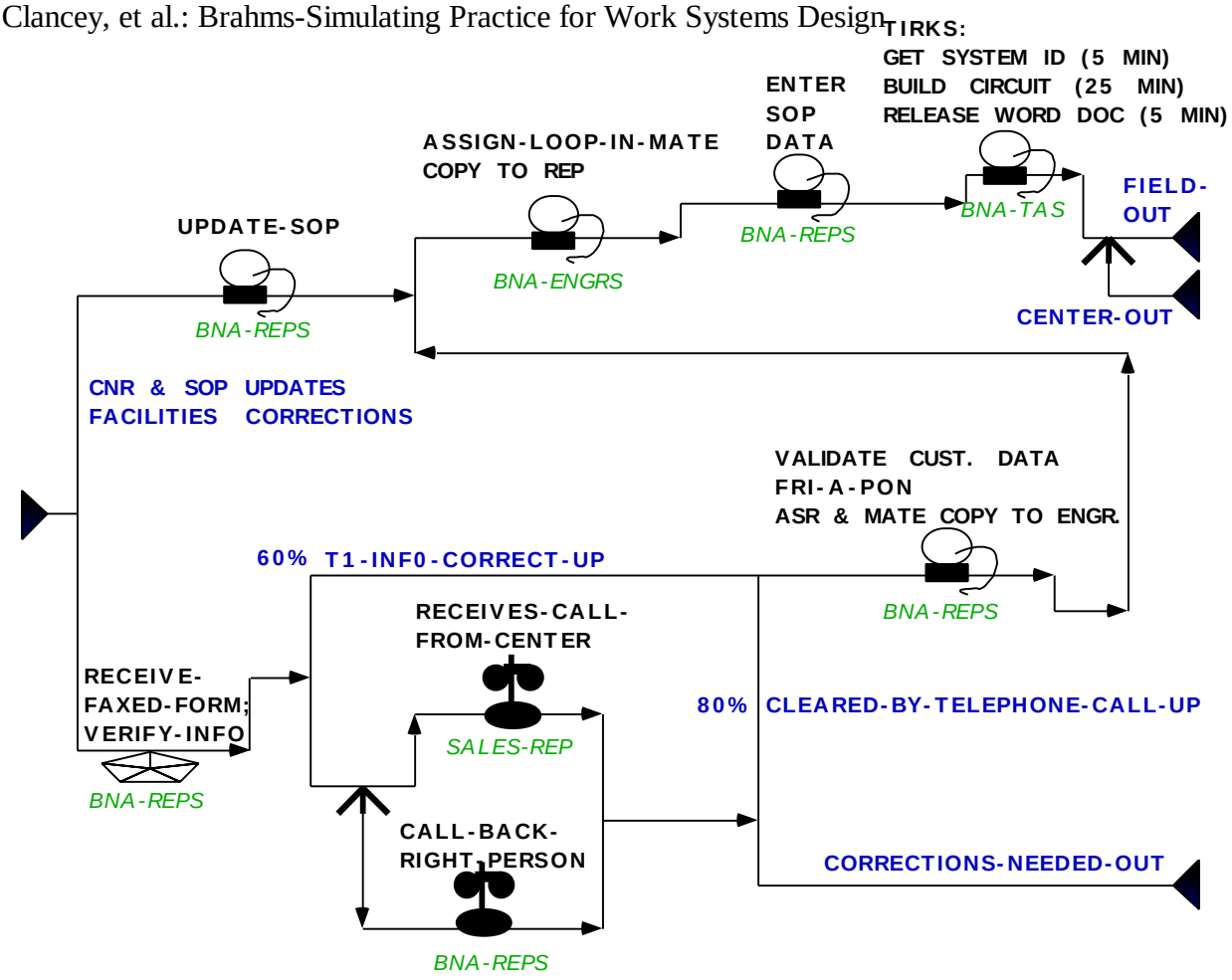
How the task view of work is incomplete

As we have indicated, business process models typically describe only idealized functions in transformation of a work product. In the following illustration (Figure 1), for example, an engineer receives an order form from a representative, assigns a circuit loop using a computer tool; later the representative enters more data about the order. A typical business process modeling tool is the SPARKS system[2] used by the Work Systems Design group at NYNEX Science and Technology[3] prior to developing Brahms. Figure 1 presents an excerpt of a SPARKS model for order processing.

A critique of this diagram from the perspective of situated action would inquire why order processing occurs this way and how it might be improved. Perhaps surprisingly, the figure leaves out what a problem solving (cognitive task) model would typically focus upon. For example, what

[2]Trademark Coopers and Lybrand. Although the authors claim that the examples presented here are representative of business process modeling, no claim is made that these are the opinions either of Coopers and Lybrand or Bell Atlantic (which subsumed NYNEX in August 1997).

[3]NYNEX merged with Bell Atlantic in 1997, after most of the work reported here was completed.



information does the engineer read from the order form and what deductions are required in order to assign the circuit loop? This particular model leaves out *how orders are planned and assigned*, *multitasking* (the fact that a rep or engineer works on several jobs at once before completing them) and *how people interrupt and resume their work* (e.g., use of notes and stacks). A cognitive model of the same business process might consider some of these factors, but would leave out how people come to be synchronized in a phone conversation, how an engineer might help a representative do his job, and broader considerations of how a representative actually spends her day. In particular, because interpreting and executing orders can be problematic in unexpected ways, people need to improvise in ways that work system designers might not have anticipated:

Figure 1. Order processing in the business network architecture (BNA) organization, showing flow of orders from left to right and conditional branching (indicated by arrows hitting a vertical bar). Top section shows updates by representatives (BNA-reps) and engineers for customer-not-read (CNR) and other revisions to orders. Lower section shows standard process for handling faxed order from sales center, followed by correcting 40% with missing or invalid information by calling the sales representative. After validating customer data (center right), orders are handled by circuit allocation process (top). Other acronyms (e.g., SOP) are internal databases.

> Information flow charts show 'information' moving in little blocks or triangles moving along arrows to encounter specific transformations and directions along the diagram. In reality, it seems, all along the arrows as well as at the nodes, that there are people helping this block to be what it needs to be—to name it, to put it under the heading where it will be seen as a recognizable variant, deciding whether to leave it in or take it out, whom to convey it to. (Wynn, 1991)

Wynn's complaint might be viewed as an issue of modeling granularity—she is asking for more details. But her broader issue is that how people think about work and how they solve problems cannot be reduced to information processing tasks and reasoning. Additional concepts are required.

To analyze the example more precisely, consider what the various branches and joins mean:

- Proportional mix of different kinds of orders, customers, or services (e.g., the first branch indicates how the order comes into the organization, as an update/correction or as an initial faxed order).
- Hand-off to the next (possibly dependent) step in a functional sequence (especially clear in the modeler's use of a step notation in the top portion of Figure 1).
- Condition of job being processed (e.g., incorrect information, indicated by a branch showing 60% correct and 40% requiring troubleshooting).
- Events that occur during troubleshooting (e.g., receive-a-call or keep-calling-back).

Although this abstraction is useful, notice that everything in this diagram was specified and connected by the modeler and workers. The model essentially leaves out the logistics, *how these conditions come to be detected and resolved*, such that work and information actually flows. What is wanted is a model that includes aspects of reasoning found in an information-processing model, plus aspects of geography, agent movement, and physical changes to the environment found in a multiagent simulation (Tokoro, 1996). The designed flow of Figure 1 assumes that people are always on the spot, picking up faxes and handing them over to others, reviewing the status of database entries on-line, responding to phone calls, etc. A designed work and information flow diagram leaves out the accomplishment of *synchronization* and the effect of *juxtaposition* of materials, such as the following:

- *Parallel-dependent processing*

  (e.g. , in practice, people start a time-consuming step in processing order before approval/clearance for doing the work at all (Dourish, MacLean, Marqvardsen, and Zbyslaw, 1996))

- *Cognitive interpretation, social knowledge, and variability*

  (e.g., how do people know that information is not correct? How are intra-group variations in how the work is done a resource for handling difficult situations, such as the unavailability of a computer system?)

- *Interactional logistics and daily activities*

  (e.g., the steps marked "Receives call" and "Call-back-right person" in Figure 1 omit the activity of "making the call" in the first place and when it occurs during the day. Is a pager and cellular phone used or voice mail at a desk phone?)

- *Informal help and "keeping an eye" on the work*

  (e.g., stepping outside defined roles, especially being concerned about the end result even after doing one's own step in a process).

By ignoring the movement and transformation of information through human action, especially conversation, a designed work/information flow not only fails to explain how flows actually can happen at all, but leaves out the effects of *serendipity*, such as stumbling on one order while looking for another or bumping into someone in the hall and learning about a new organizational priority.

A coffee meeting: Interaction of identity and computer tools

As an example of how Brahms modeling brings out circumstantial social-interactional factors, consider the following incident. We had constructed a preliminary model of the practice of order processing as it occurs among wirers and other technicians subsequent to BNA order processing. In this model, we included a "coffee meeting" activity for service technicians and their field supervisor lasting from 8 AM until 8:20 AM. We showed this model to our colleague and collaborator, an operations specialist from NYNEX. He responded, "You can't show this to management!" We objected that we wanted to represent what people actually did during the day, and a coffee meeting had to be included. He indicated that including the meeting wasn't the problem; we had not shown what occurred during the coffee meeting, namely that the supervisor communicates to the service technicians what jobs are prioritized for the day and their particular assignments. This was news to us and valuable information, for it added a key piece to the model, namely how service technicians found out about new orders. Not wanting to make assumptions especially about whether and how work was planned, we had initially left out this aspect of the model. Mentioning this to our collaborator we then learned that assignments are strongly biased by the supervisor's knowledge of the knowledge of the service technicians, specifically whether the

technician could be trusted to satisfy a particular client. We call this *social knowledge*. In this respect, planning orders is socially situated, influenced by who is available, who is asking for a new job at a particular time, and what people know about each other's capability. (This illustrates one aspect of the *knowledge content* analysis of a situated cognition perspective, mentioned before.)

At this point in our conversation with our collaborator, we generalized the apparent social interaction and deduced that the switching electronic technicians (SETs) and their supervisor, the "turf coordinator," have the same practice in the central office. But it turned out that the SETs find out about new orders and their daily assignments after the coffee meeting by examining the schedule previously posted on-line by the turf coordinator. Why doesn't this group have the same practice as the technicians out in the field? Why do they prefer to use an indirect approach involving a computer system? In fact, the turf coordinator is another SET on temporary assignment as a supervisor. One SET would not ordinarily tell another what to do. Thus, a proper "social" way of coordinating their work is to make and communicate assignments indirectly, using a computer system.

The example illustrates how understanding practice, including whether and when people will use computer tools, requires relating multiple perspectives. Adopting an idealized, optimizing view, one might have suggested that the coffee meeting was a distraction and that all groups should handle assignments through an on-line system. After all, an on-line system would be available from many locations, provides a written record, is easily changed, etc. But this would ignore the other aspects of planning and communication about past and future orders that occurs in the coffee meeting. Eliminating the coffee meeting would eliminate how information is informally shared and people are learning about other jobs, methods, and each other. On the other hand, one must not adopt an idealized view about social interactions either. Concluding that all planning should occur during face-to-face interaction, such as at a coffee meeting, would ignore how a computer tool can provide a resource for handling a key issue of social identity, namely that in some groups it is not permissible for peers to tell each other what to do.

Thus, we have found in this simple coffee meeting an example of how two groups accomplish the same "task" of order assignment in strikingly different ways, based on the constraints and opportunities afforded by tools and social relations. The example illustrates how the turf coordinator's identity with respect to the SET community of practice makes a computer solution more tenable, although it may have other disadvantages over a face-to-face discussion. How these groups accomplish their work cannot be easily predicted either from a technical or a social perspective alone. The organization, processes, and tools cannot be effectively changed without a systemic view of diverse, interacting social, physical, and psychological influences. The example

illustrates how redesigning a "human-computer system" is facilitated by relating these different views.

In view of these considerations, we deliberately chose to include multiple, complementary views in Brahms, rather than the single-dimensional perspective of the work flow model (as shown in Figure 1). Specifically, people performing the same task may do so in different ways based on the group to which they belong, not just for historical reasons, but because of their interpersonal relations to each other. Logically speaking, there is no reason why SETs can't tell each other what to do. But human actions are not just technical, functional performances, but indirectly, even when we might prefer otherwise, are conceived as constructing *and maintaining* an identity relative to other people. The social effects of actions necessarily constrain choices people make about how to accomplish their formal job functions.

A dynamic model: Local pieces interact to produce flows

Based on analyses like that presented of the coffee meeting, we decided that Brahms should model the activities of individual agents who are physically located[4] (so they can be part of face-to-face meetings or not) and behave according to their membership in different groups (which includes informal identification, i.e., a BNA-rep who is trying to become an engineer). What results is something between the task analysis of business process and cognitive modeling—a Brahms model includes not just tasks or knowledge, but activities, ways in which people *chunk their daily life to scope their attention and concern.* Common office activities include coffee meetings, reading one's mail, answering phone messages, being a supervisor on temporary assignment, etc. This is the "background" of practices mentioned by Wynn, which is ordinarily not included in models of work processes. In short, Brahms models are not necessarily as detailed as models of cognitive skills (though a modeler could choose to do this) nor are they as general as functional models of business processes (Figure 2).

[4]Recall that expert systems model information processing (aka *reasoning*), not *behavior* of agents per se. For example, a medical expert system typically models disease diagnosis and therapy planning, not a face-to-face conversation with a patient in a particular kind of room or what medical equipment is used.

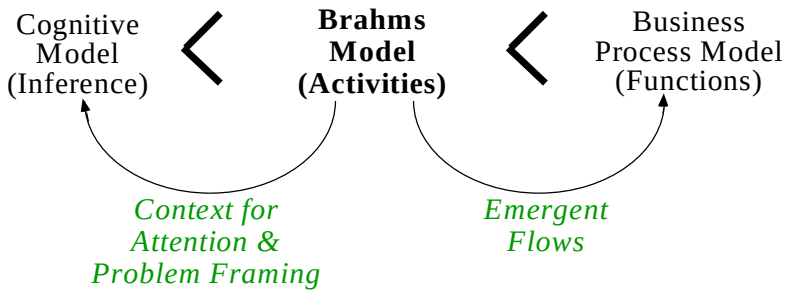


Figure 2. Level of abstraction of Brahms relative to other models of work. Activities are more abstract than inference and less abstract than business functions.

A Brahms model does not necessarily describe the intricate details of reasoning or calculation, but instead models the social-physical context in which reasoning occurs, especially how observations are made based on what activities the person is already doing (such that a problem situation may be detected). A Brahms model does not describe only what people are supposed to accomplish (e.g., functional transformations of materials). Instead, such diagrams are derived from the results of the interactions that occur; that is workflows are emergent during the simulation.

To be sure, any qualitative model will specify conditions and consequences in a rule-like way. We still build into Brahms simplified descriptions, which are idealizations. But first, they are descriptions of *spatially and temporally located behavior*, and second they are *smaller pieces*, such as the many ways in which information and work product paths may be formed. That is, the model provides for circumstantial interactions to occur and to have an effect. In this sense, because the flows are pieced together dynamically as the model runs, the flows are emergent. The modeler is still of course concerned with overall completeness and connectivity of the model, but the relevant factors must be specified in terms of the informal, circumstantial influences by which a practice develops. Due to interactions between agents, physical objects, and locations, the paths followed and the quality of results will be contingent on what resources are actually available, who actually participates, etc.

As an example, consider how phone conversations are modeled in different simulation tools. In Sparks, one can leave out entirely that a phone call is even made, focusing on the information conveyed. The problem of synchronization (having two people at the same time in a conversation) is handled stocastically, by modeling the time required for information transfer as a statistical function (e.g., 80% of the time a conversation occurs within an hour, but 20% of the time it will take a full day). In another multiagent simulation such as Virtual Design Team (Levitt, Jin, Oralkan., Kunz, and Christiansen, 1995), a phone conversation is modeled as a message arriving in an in-box, which disappears if it isn't read within a minute. How the caller knows that the message was not received is not considered. In Brahms, phone calls are modeled in considerably more detail including phone numbers, busy signals, hearing a phone ringing, deciding that the

conversation is over, etc. Thus, whether the conversation occurs is not built in as a statistic, but is the emergent result of where the agents and phones are located, what people are doing when a phone rings, etc. In contrast, a model such as ITHINK[5] is also dynamic, but generally does not represent individual agents, phones, locations, etc. Rather, ITHINK represents aggregate behavior of a group, representing not particular actions at particular moments by particular agents, but the cumulative influences of an organization, leading to the total number of orders processed, the total number of backlogged orders, etc.

Finally, in designing Brahms, we concluded that we should be flexible in how we create and use models. For example, one can begin modeling by pretending that one agent does all the work (a useful heuristic for identifying what tasks are essential), or model multiple groups with one agent each, or place all the agents in one location or move them all apart, etc. We view model building as inherently an experimental endeavor, aimed at gaining insights, and only secondarily, if at all, in developing something "complete" or making accurate predictions. Our predominant interest is to create models in order, first, to engage workers in work systems design conversations and, second, as a tool for raising good questions (often from the obvious deficiencies of the model, as in the example of the coffee meeting). On the other hand, models may be evaluated in terms of statistics of flows generated during a simulation run (e.g., number of orders processed/day/person or group, backlogs, number and kinds of errors generated/detected/resolved, touch-time/job). But the *presentational* value of the computer model—like a theatrical play—is our top-most concern, using Brahms as a way of representing complex human-computer systems in a way that helps people reflect on their practices and how to improve them.

A theatrical presentation, not necessarily a scientific model

It should now be clear that although we have said that Brahms represents conditional actions as constructs that resemble rules, these are unlike rules in a knowledge-based system. First, the constructs describe behaviors (*workframes*), as well as inferences (*thoughtframes*). In contrast with most knowledge acquisition efforts, which strive to build a universally applicable model ("knowledge base"), Brahms modeling is deliberately partial and aimed at illustrating issues relevant to an on-going work systems design effort. In contrast with aiming for terms and definitions that are (supposedly) uniformly interpreted by every reader, Brahms modeling assumes that people will interpret models differently, depending on their identity, interests, and values. In fact, this is why we build such models, to facilitate people learning about each others' perspectives and ways of talking about what is analytically just a task flow. (Consider for example what you,

[5]Trademark, High Performance Systems.

the reader, have now learned about how SETs and Service Technicians approach something so simple as daily job assignments.)

The behaviors we model in Brahms include reasoning, but more generally include ways of coordinating with other jobs/tasks, stepping outside literal responsibility to offer assistance, and a host of general activities such as "reading my e-mail," "following the service technician to the customer's site in my truck," and "having coffee with my supervisor." A model of activities *includes* traditional components of cognitive modeling (specifically, the vocabulary of "belief" and "inference"), Traditional problem-solving models are embedded in this larger simulation of activities, not replaced. Similarly, as stated, business process workflow diagrams are derived from a Brahms simulation. They are not replaced by Brahms, but rather generated in a different way.

Some researchers view every model that includes inferential processes as being a *mechanism* of how the brain works or how cognition occurs in the individual. But aside from coarsely mimicking behaviors of people, nothing in Brahms requires such a commitment. Brahms is not designed to replace people or replicate their intelligence, per se, no more than a play in a book replaces people and their experiences. A Brahms model, like a play, is just a kind of map, not the territory.

From the perspective of work systems design, Brahms models should be evocative and have a basis in reality. But to be useful, a Brahms model need not replicate or predict any particular group's behavior. Keeping in mind that the audience of workers often consists of people without advanced university degrees, a dramatical presentation is a way of making complex theories about human interaction accessible. A model may have rhetorical effect, bringing about an emotional response — "No, it's not like that! We do work doing the coffee meeting, we don't just sit around and talk," or maybe, "Yes, that's exactly what is going on here at Pearl Street—can we show this to people at the World Trade Center so they can see what things are like over here?" On the other hand, one might devote sufficient effort to portions of the model so it will have enough veracity to allow testing hypotheses about various redesigns and making relative numerical comparisons (e.g., time and cost for processing orders).

In summary, Brahms's architecture is partly a reconception of the meaning and use of existing modeling techniques. In this respect, a Brahms model could be viewed as a group of interacting knowledge-based systems in a simulated environment. But new architectural features were required for modeling how people conceive of activity and how attention and inference are contextually scoped by parallel-ongoing activities (described below).

TECHNICAL APPROACH AND ARCHITECTURE

In this section we describe the technical approach of multiagent simulation models, the requirements specification for Brahms, and the representational architecture.

Multiagent simulations of work

Several simulation tools enable modeling organizational command and coordination policies in a geographically distributed environment, in a way that broadly fits a model of practice. For example, Cohen, Greenberg, Hart, and Howe's (1989) Phoenix simulation models coordination of fire-fighting teams. Hayes-Roth, Brownston, and Sincoff's (1995) simulations allow for improvisation in games played by the agents. Tambe, Johnson, Jones, Laird, Rosenbloom, and Schwamb (1995) describe a tool that models social interactions such as briefing sessions before military missions. In general, such multiagent simulation tools have the following characteristics:

- *Agents* are modeled as active, controlling the work flow by their beliefs, decisions & actions, not as resources passively assigned to work; agent behavior is asynchronous. Each agent is a "knowledge-based system" that monitors and controls its own behavior.
- That is, the model is based on *located, interactive behaviors of objects and agents,* not just abstracted (disembodied) process of work flow.
- *Attention* of actors at each moment in time is modeled explicitly.
- The *environment* is explicitly modeled, with its own objects, states & causal processes.
- In ecological models (e.g., SimLife^{TM}) *all interactions are transactions* (because environment triggering actions is itself changing and being changed by actions).
- *Communications* between agents are modeled, which change beliefs and trigger new actions.
- Reciprocal information requirements between concurrent activities are modeled.
- *Movements* of agents are modeled.
- Agent behavior is modeled as a *problem-space (task) hierarchy*
- An *organization* model indicates work capabilities and controlling privileges (e.g., task assignments)
- *Inferences* (decision making) of agents are modeled by production rules.
- *Throughput of the work group emerges through interactive behaviors* of all agents, not modeled as equations based on numeric parameters describing the group (e.g., "amount of knowledge").

As we have indicated, these tools provide an overall framework for developing a model of practice. In particular, models of business "enterprises" represent the complex coordination between agents playing different roles. For example, Levitt, et al.'s (1995) VDT models both

inefficient behavior as well as idealized "intelligent" behavior. Burstein, Ferguson, and Abrett (1993) describe a tool for designing group coordination strategies for efficiency at peak loads, using "coordination structures" as templates for creating models, such as a template for an administrator who supervises agents who are clones. Dozens of programs focus specifically on modeling animal, computational, and even early human societies (e.g., see Gustavsson, 1993; Gilbert and Doran, 1993).

These systems have the following characteristics:

- They emphasize *event-triggered interdependencies*, e.g., when an order arrives on a fax, a person in a given role processes the order.
- They *de-emphasize or ignore work practice*, including ambiguity, exceptions, change, visuals, creative improvisation, redundancy, "out louds" (speaking into a room for anyone to hear)
- They do not represent how *synchronization* (and hence coordination) of agent behavior occurs through their actions (such as in walking over to someone and saying, "I need to talk to you")
- They tend to be concerned with the *technical details of reasoning* (e.g., the calculations involved in assigning telecommunication circuits), rather than how a particular person came to be involved at all.
- They do not explicitly describe the *activities* by which people organize their tasks, by which their attention and interactions with the world are chunked and scoped.

Such tools are sometimes more suitable for modeling clerical forms processing (in which kinds of information used and roles are relatively stable), than for analyzing design and planning tasks (in which the information and team are dynamically constructed).

In general, the notion of "social" in simulations of organizations is quite impoverished. For example, one tool models social behaviors in terms of "decreasing information-processing capability" (e.g., emotional responses) and deceit (Carley, Park, and Prietula, 1993, p. 3). Malone and Crowston (1991) define coordination as the "act of working together" but apply the conventional management perspective. Their models do not capture practice, but instead descriptively *abstract* coordination in terms of bidding and communicating interdependencies. Indeed, they view coordination as "the additional information processing performed when multiple, connected actors pursue goals that a single actor pursuing the same goals would not perform." That is, coordination is the *overhead* required when you can't do everything yourself!

The next subsection summarizes the specification that distinguishes Brahms from other multiagent simulations.

Specification for the modeling language and simulation engine

In the preceding discussion, several design requirements for Brahms were mentioned:

- *Model chronological behaviors (activities) of multiple agents* — a representational architecture that allows for *agents* belonging to multiple *groups*, located and moving through some *geography*, which includes active artifacts such as telephones and fax machines.
- *Model the reactive nature of human behavior* — how people dynamically shift their attention, interrupt and resume activities, and compose activities (having multiple interests at once).
- *Model conversations* at the level of sequences of ask/tell interactions, reading and writing documents and databases.
- *Model how a conversational interaction becomes synchronized* — a protocol for face-to-face conversations and a protocol by which telephones are physically connected and used.
- *Model how information is represented, transformed, reinterpreted in various physical modalities* — fax machines, databases, forms, multiple pass reviewing and reading, location and movement of documents (e.g., tables posted on walls in the office).
- *Produce multiple graphic views of work* (e.g., geographic, agent-centric (chronological), order-centric (workflow)) amenable for use by workers and managers, especially across organizations.
    - Model the work (task) flow view by generating it from local behaviors that occur during a simulation run.
    - Model the "cost" and time — "touch time," such as how many people spent how long working on a given order, basic statistics about order processing that would be generated by a business process model.

Design of Brahms began in early 1993. The architecture was inspired by the hierarchical, abstract task structures of Neomycin's diagnostic procedure (Clancey, 1992), the subsumption architecture of Brooks' (1991) robots, and the distributed processing architectures found in artificial life simulations, such as the graphic multiagent simulations developed by Maxis for SimLife^{TM} and SimCity^{TM}. Briefly, activities are represented as conditional actions called workframes, and are

inherited from the groups to which the agent belongs. Workframes may activate other activities, such that a hierarchy of activities is simultaneously active, as in the subsumption architecture, modeling how a person's conceptualizing is coordinating many on-going activities at different grainsizes. Someone not understanding the difference between tasks and activities might believe that the representational language does not work as expected. Normally, procedures on a stack "invoke" subprocesses, rather than continuing to operate in parallel, and "return" products, rather than sustaining interactive behaviors. Subsequent sections provide further details.

Representation language details

The most central representational unit in Brahms is called a *workframe* (Figure 3), a situation-action rule consisting of preconditions (what the agent must believe to be true), actions, detectables (what facts in the world might be noticed, with what probability and when during the actions), and consequences (changes to the world or this agent's beliefs that result). The example shown is a workframe for Field Supervisors. It is at the "top level," which means that it is always potentially activated by members of this group. The workframe is activated after 8 AM; the effect is that the Field Supervisor will notice every service technician in the room and engage in "Morning Planning," an activity.

Workframes are organized hierarchically into activities (e.g., Morning Planning with STs is an activity with one workframe, shown in Figure 4). Actions in a workframe may be simple (just indicating a name, duration, and priority) or composite (another activity). Figure 4 indicates that during the Morning Planning activity, the Field Supervisor engages in face-to-face conversation with service technicians. For each service tech and order that needs to be discussed, the Field Supervisor will tell the service tech that he or she is assigned to an order.

Simple actions also include movement to another location. Consequences and actions are ordered and interleaved. Detectables may be indicated as "impasses" that interrupt the workframe or as "end conditions" that end the workframe or its encompassing activity. The detectable in Figure 3 simply observes facts in the environment.

WORKFRAME: Morning Coffee Meeting

VARIABLES:
ST = an Agent

PRE-CONDITIONS:
Agent knows: The hour of the clock is >= 8
Agent knows: The group membership of **ST** is Service Technician

SIMPLE ACTION: Notice who's in the room (5 minutes)
DETECTABLE: The on-job-status of **ST** is present.

COMPOSITE ACTION: Morning Planning with STs

Figure 3. Example of a workframe invoking an activity, written informally.

WORKFRAME: Face to Face Conversation with Service Technician about Assignment

VARIABLES:
Cimap-Order = an Order
ST = an Agent

PRE-CONDITIONS:
Agent knows: **ST** needs to be told about **Cimap-Order**
Agent knows: **Current Actor** is communicating with **ST**

SIMPLE ACTION: Talk (5 minutes)
COMMUNICATION ACTION: Transfer to **ST**: The service technician assigned to **Cimap-Order** is **ST**

CONSEQUENCES:
ST needs to be told about **Cimap-Order** is FALSE
Current Actor needs to talk to **ST** is FALSE

Figure 4. Example of a workframe with a communication action.

Workframes are inherited by agents from all groups to which they belong; groups may belong to other groups (Figure 5). Priorities allow workframes to interrupt each other or carry out specific aspects of a more general protocol. For example, workframes at the "all groups" (top) level

specify how to use a telephone and have face-to-face conversations; these have intermediate priority. Workframes that trigger conversations are most specific and have the lowest priority. Workframes that specify what to say during certain kinds of conversations have the highest priority. By this simple scheme, it is possible for one agent to initiate a conversation and for the responder to "remember" something he wanted to tell the first agent when he called; thus a give and take may ensue.

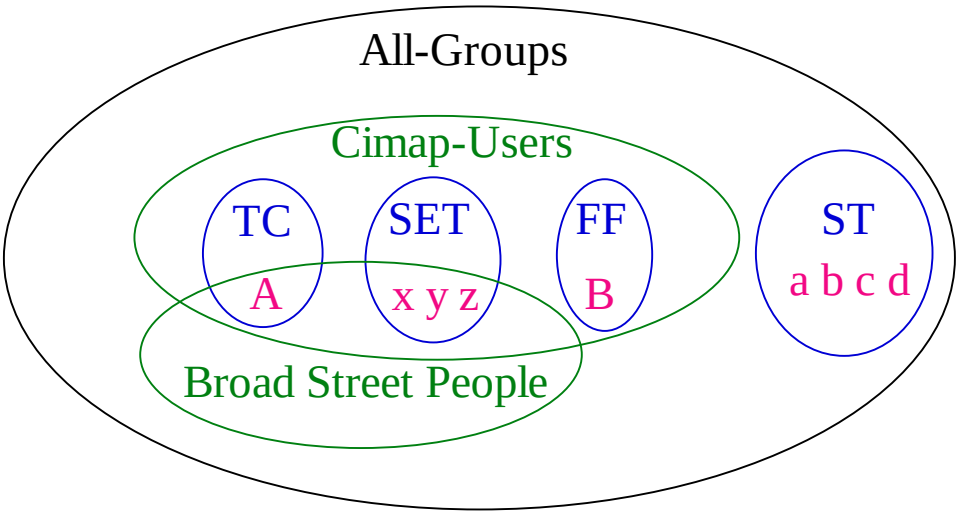


Figure 5. Groups definitions include initial beliefs (e.g., social knowledge about other's capabilities), behaviors in using technology and representational media, and daily activities. As shown here, agent A is a Turf Coordinator who works at Broad Street and uses the Cimap database.

Thoughtframes model agent reasoning about implications of beliefs, leading to changes in what they do next (thus a distinction is draw between "action rules" and "thinking rules") Thoughtframes take no time.

Changes to beliefs may occur by virtue of: broadcast (e.g., speaking outloud), transfer from agent (telling or asking), transfer from object (e.g., reading a database or a fax), detectables, and consequences.

Activities are spatially-dependent:

- *location goals* cause the agent to move to location when workframe is enabled (e.g., "Move to location X.")
- *location preconditions* depend on agent location (e.g., "Is the current agent at location X?")

Objects embody stored information about the world, modeled as the "beliefs" of the object (e.g., a database). *Factframes* models object behavior, including what they detect and how they change

state. Object instances may be created by an action (e.g., fax transmission creates a paper copy at the receiving station).

Facts are an eagle-eye view-from-nowhere—the outsider's view of the simulation, for example, the state of telephones, location of agents, etc. Detectables specify what facts an agent might detect during the action of a workframe. Beliefs are propositions agents believe about objects (state of the world) or other agents.

A communication may involve asking or telling. A communication may be from an agent or object to a specific agent or object, a group of agents, a class of objects, or may be broadcast. For example, a factframe for the fax object broadcasts to every agent within proximity that a fax has arrived. An agent can only communicate what he or she believes.

Brahms currently models geography in a rudimentary way, consisting of regions, buildings, and their connections. Duration of movement is simply proportional to distance; for convenience movement between non-connected locations takes no time. We believe that our objective of making models accessible will only be realized when graphics are incorporated of the caliber one commonly finds in games on personal computers.

In general, descriptions of activities are associated with groups. In practice, there may only be one member of a group in a given workplace (e.g., one “customer representative” at the customer’s site) or roles may be highly differentiated (e.g., the role of the “turf coordinator”). Depending on the purpose for building the model, models may represent:

- *particular people* (Madeline at the BNA Center),
- *types of people* (“a BNA engineer at Pearl Street”), or
- *pastiches* (“a typical customer representative, patterned after Marilyn and the lady in the purple hat”).

Agents that are not central to the work being modeled may be modeled as an individual representing a group. For example, an aggregate “customer” for a workgroup could generate orders. Like a play, only the main character (in our case the Turf Coordinator) would be modeled in detail, such that if the agent is shown to be idle during the simulation something is known to be missing. Other agents are like characters in a play who come and go from the stage, such that it may not matter if they are idle when they are not interacting with the main character(s).

RESULTS

Two detailed, connected models have been created: front-end order processing in the NYNEX BNA Center and the back-end circuit wiring and testing of the technicians, splicers, etc. The back-

end ("turf coordinator") model was created originally as part of specifying requirements for Brahms. The problem was chosen because testing a circuit occurs when three agents are in a conference call at three locations—a form of synchronization that people find demanding and that we knew would challenge our modeling skills. Analyzing existing Sparks models enabled us over the course of several years to articulate the difference between task and activity models (Clancey, 1997b) and how a workflow diagrams could be generated automatically. Building this model also revealed how planning occurs on the job (the coffee meeting example). We deliberately modeled as much at the "all-group" level as possible (Figure 5), to make the model adaptable for other settings; components such as the phone and fax machine models are directly reusable.

The front-end model was begun in 1996 and was intended to help managers and software engineers understand why orders rejected by an on-line system were generated and resolved. Workers collaborating with us found the modeling process to be valuable. Specifically, the focus on what people actually did when they processed orders revealed that the program's rejects, called "errors" heretofore, were not necessarily human mistakes, but just orders that the software could not process. Our systemic approach led backwards from this downstream processing (in another part of Manhattan) back to the BNA Center and to its peer at World Trade Center. The actual causes of computer system problems were found to be not just typos, but primarily an inability to specify certain kinds of jobs using existing forms. Assumptions built into software also ignored pragmatic issues, such as the need to start order processing before getting customer credit approval. We also showed that the error rate dropped not because of "fewer mistakes," but because the work group shifted to a fully manual process that worked around the limitations of the order-processing that was supposed to partially automate circuit design. This analysis raised each group's awareness of the other's work and gave the software engineer responsible for the downstream system a better appreciation of the difficulties the BNA Center encountered and appropriately handled.

Finally, in the back-end turf coordinator model we had treated all members of a work group as being clones, such that all SETs behave identically. This choice followed from the social science preference not to view people as individuals, but to focus on trends and commonalities. In building this second model we questioned this simplification and asked how the individuals in a group differed from one another. Our analysis of BNA engineers revealed a kind of knowledge variability that was unexpected—people were of course not clones, but differences were not errors, either. We found alternative methods being used for the same task (such as verifying that a given circuit was available for assignment); we conjecture that such differences are a vital source for learning. Possibly the lack of consideration of individual differences heretofore by the social scientists led them to emphasize cross-functional learning (one group learning to do another's tasks), rather than

learning among people with similar responsibilities. Furthermore, the idea of legitimate knowledge variability contrasts with the typical corporate view that all variability in job performance is non-optimal or based on misconceptions or lack of knowledge (hence, one goal of corporate training is standardization). Instead, knowledge variability may be a source of robustness, allowing adaptability when the environment changes (like variability in a biological population).

Activity-based modeling in Brahms led us to ask new kinds of questions about the BNA Center:

- How flows occur: How do reps coordinate completion of orders with engineers?
- End-to-End systemic perspective: Does the circuit assignment system ensure that billing occurs and that there is adequate inventory?
- Knowledge variability: How does knowledge of circuit technologies vary among reps and engineers in the center?
- Concern about different practices (for the same process): What's actually happening in the National Sales Office at the World Trade Center, where two reps are doing an engineer's job?
- Effect of work location on the quality of work: How is the new intermingling of services (combinations of circuit types) relevant to co-location of order processing in one center?
- Relation of different representational media: How is *Lotus Notes* used to fill in gaps between legacy systems?

In summary, the Brahms approach of considering agents interacting in a social-physical environment facilitated understanding a human-computer system whose complexity transcends the normal range of task/cognitive modeling. The analysis revealed and gave legitimacy to informal behaviors, such as a coffee meeting, which are typically omitted from business process models such as Sparks (Figure 1).

To understand the advantages of activity-based modeling for developing a software agent application, such as an "intelligent agent," consider the example shown in Figure 5.

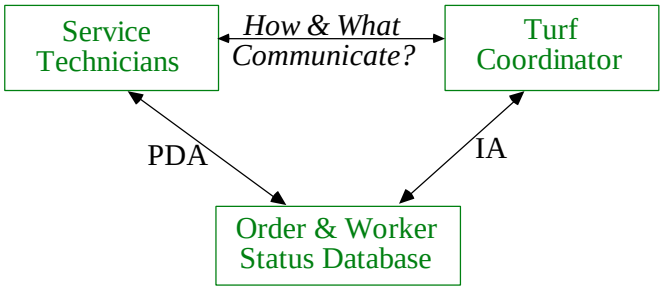


Figure 5. Given technicians out in the field communicating the status of their work back to a central database via a personal digital device, how might an intelligent software agent help the turf coordinator know what is happening in the field and what advice might it offer for prioritizing and making job assignments?

Given information about the location of different service technicians and knowing that the turf coordinator had last read the order database that morning and wouldn't review it again until the following morning, a program might offer advice such as, "TC Allen, ST Aronson just completed the job down on Wall Street and is now available; perhaps you want to have her go over to Broadway to handle the Teleport job?" More broadly, activity-based modeling provides a new way of modeling users (such as the turf coordinator), which includes not only what tasks they do and the information they use, plus some of the deductions they might make, but also where and when they do such reasoning, where they might be found at a particular moment, who might know where they are located, what interests them at a particular time of day, etc. *Thus a Brahms user model would combine cognitive and social-interactional considerations.* Similarly, such a model would be potentially more useful for instruction than a typical knowledge-based model because it would help a student understand the practices by which different tools are related, who is typically available for providing help, what kind of assistance may be sought, and so on. Finally, one could use Brahms for implementing a software agent itself, locating the agent in the modeled social-interactional context, making explicit what external resources are available to it, how it should behave when participating in different groups, what it should do at different times of the day, and so on. In summary, activity-based modeling provides a way to inform computer systems of everyday practice of the people they are serving, and thus, in a very limited way, integrate them into human communities.

DISCUSSION

In this section we elaborate on some theoretical issues concerning the relation of Brahms modeling to knowledge engineering, what Brahms models omit, and how our research changes how we view the relation between anthropology and computer science.

Different epistemological stance than knowledge engineering

Brahms is designed primarily for modeling what people *do*, rather than what they know. We focus on a work group's choreography, not just an individual's reasoning. In effect, existing cognitive models ignore the effect of interactional pragmatics on problem solving and make overly strong claims about what knowledge is shared in a group:

- *Individual capacity* (such as the ability to detect and resolve trouble) *is a product of individual habits and environmental processes* (including especially interpersonal relations). That is, problem solving behavior—what information and methods are used and hence the quality of the work—is not strictly controlled by the individual's reasoning, but is the result of an interaction between internal (in the head) and external (in the world) influences (Hutchins, 1995).
- Observed patterns of activity in a work place are emergent, *not scripts* that are shared (literally known) by everyone in a group. For example, there is no such thing as one restaurant script that everyone knows. There are waiter activities, chef activities, cashier activities, patron activities, etc. If these activities are properly choreographed, it appears as if the team knows what the other agents are doing. But in fact much of what other agents do is invisible to each agent, and proper coordination is the result of local interactions, not global knowledge of "the script."

Human knowledge is situated in the sense of being subjective (formulated and evaluated with respect to a social role) and inherently interactional (oriented towards how to behave, including especially social norms), not a universal view from nowhere (Nagel, 1986). Consequently, many descriptions of knowledge in AI models capture what a group accomplishes (product-oriented descriptions of object transformations), which constitutes a kind of scientific model (e.g., how electronic circuits function and how symptoms relate to circuit faults), not what individuals know about each other and how to behave intelligently.

The demands of modeling work practice almost turn inside out the conventional view of knowledge engineering. Context is not just something in the environment ("data"), but partly *conceptual* and partly about other people. A social system is not just an organization, but *a choreography of interaction*, a set of practices for doing things in certain places at certain times. Knowledge is not just technical, but is about the group—*social knowledge*. What people know and do is organized around their roles as social actors; they are not plug-compatible problem solvers, but people with different interests and different ways of working together. A kind of task might be accomplished pragmatically in many ways, without the variability being a matter of misconceptions or missing knowledge. Expertise includes knowing what other people know, how

to get help, who is trustworthy and who is diplomatic, and how to team a patient, careful worker with an imaginative explorer.

In modeling work practice, standard AI issues of scheduling, planning, and information processing are not omitted, but are made *problematic*: How does a supervisor remember what everyone is doing during the day? How do members of a team at different locations coordinate their work day? How does informal, circumstantial encounters (such as conversations in a hallway) help align the expectations and understanding of the group about group's capabilities, how busy they are, and what they are becoming? To say that such issues are ignored in expert systems is an understatement. Indeed, social knowledge and interaction are ignored in most software engineering tools for designing computer systems. But how can we design computer tools if we don't know what people need? That's the ultimate value of modeling work practice.

This much said, we hasten to point out that Brahms' models of practice are exceedingly limited. One value of developing Brahms has been to highlight for us the concerns that are typically raised in work system design that are not easily modeled in the current Brahms language. In particular, Brahms models do not represent the following:

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

We could model conversations in more detail (at the cost of a much slower simulation) by incorporating models from conversational analysis; one reason for doing this is to make explicitly how social-interactional influences are manifest in low-level conversational practices, such as turn-taking. In related work, Whalen and Vinkhuyzen (forthcoming) show that such analysis is relevant for designing a computer tool to be used by a service representative talking on the phone with a customer.

Similarly, we believe that cognitive methods for modeling could be embedded in Brahms, with a resulting model that is more comprehensive and insightful about how learning actually occurs. This is of particular interest for analyzing learning on the job (Clancey, 1993).

Other aspects omitted involve historical, cumulative effects, which are considered by a variety of disciplines ranging from sociology to human factors. On the one hand, the sociological theories would benefit from being forced to relate to contingent psychological and physical factors; the human factors theories would need to relate to a social analysis of norms and values.

Modeling reconceptualization and juggling of activities requires better neuropsychological theories of memory, attention, and perception. The use of the subsumption architecture in Brahms for modeling conceptual hierarchies (of activities) may be usefully extended by combination with a selectionist or connectionist model of multiple constraint satisfaction (e.g., Mitchell, 1993).

Finally, broader considerations such as life away from work may be directly incorporated by extending the models we have developed to include families, weekend activities, and so on. This would be especially useful if one wanted to analyze and represent what workers learn about each other through informal activities outside of work. As for all of the considerations listed here and those already included in Brahms models, the relevance of any level of detail depends on the kind of work system being designed and pragmatics of the design process.

The design stance in "business anthropology"

Brahms was developed through a collaboration of anthropologists and computer scientists. We believe that both of our research disciplines are changed by developing a tool that enables us to work together. On the one hand, computer science brings a design stance to anthropology, on the other hand, as emphasized throughout, anthropology brings an understanding of the context of use to software engineering.

A work systems design effort presupposes what engineers call a "design stance." That is, rather than simply studying or describing a workplace, which has been the conventional approach in anthropology, we intend to use Brahms to change how work gets done. This means that we must have theories of *how different kinds of designs produce different results*, otherwise we would not be able to suggest alternative designs or reason about trade-offs. Of course, the very idea of participatory design means that facilitators (e.g., anthropologists and model builders) need not (and could not) develop an optimal design and deliver it to clients. Instead, Brahms is conceived as a tool by which researcher-facilitators in collaboration with workers and management will bring about change incrementally and iteratively. Nevertheless, using Brahms as a design tool presupposes that facilitators have some understanding of how interactions between people, technology, and organizations relate to *measurable changes in work quality*. In contrast with

traditional ethnographic descriptions of cultures, observations and models of work practice are fundamentally *evaluative*.

For example, to show the benefits of alternative work system designs, we might model how beliefs people have about members of their local work group (e.g., who is doing what with what results) change because of the design of work processes and the workplace layout. For instance, co-location (working in the same office) facilitates face-to-face conversations, in which beliefs about the group are communicated (e.g., service technicians during a coffee meeting might learn about how a given technique produced an expensive failure at a customer site; Orr, 1995). In contrast to communications via on-line systems, face-to-face conversations tend to articulate *beliefs about skills and experience*, not just beliefs about the status of different orders. We can show these relations causally:

design of facilities, technology, & organization
=causes=> *communication of beliefs about the group*
=causes=> *changes in how people participate*
(e.g., assigning jobs, getting help, dividing up the work, teaming people)

Interactions between designs, beliefs, and work are crucial from a cognitive perspective, too, for *how participation is determined* will affect what knowledge is brought to bear on a problem. As indicated in the previous subsection, a more complete model would of course include a model of how skills change and how learning is influenced by different work system designs.

A simple way of stating our incipient understanding of work systems design is given by this causal relation:

design -> interactions -> workflow -> results

A given *design* (configuration of people, technology, and organization) causes certain everyday *interactions* in activities (both positive and negative, including antagonistic relationships and customer-focused conversations). These interactions are manifest in certain properties of the *workflow* (e.g., bottlenecks, early detection of errors, multiple error-prone hand-offs). Finally, these workflow properties are manifest in the *business results* management uses to assess productivity (e.g., average number of days to process an order).

As an example, consider the logic behind part of the redesign of the back-end order provisioning process (Figure 5).

Work System Design: co-location, turf coordinator, turf teams

—>**Activity Interactions:** improved communication (trading contextual information), collaborative planning & troubleshooting

—>**Workflow Quality:** bridges, hand-offs, no silos

—>**Business Results:** fewer errors, shortened interval (CD)

Figure 5: Relation of design to business results in order provisioning

The design incorporated elements such as "co-location" and a turf coordinator. This combination of people, technology, and organization changed the interactions occurring during work activities to improve communication and promote collaborative planning. These changes in turn affected the workflow: Bridging front and back ends of order processing and reduced hand-offs released work early, provided a single dispatch to the customer, made scheduling of jobs more efficient, etc. This in turn lowered the error rate and shortened the time required to turn up the circuit.

This example illustrates that reasoning about designs requires adopting different perspectives and arguing about how they are *causally related*. The advantage of the ethnographic perspective is that it introduces the level of "activity interactions," complementing the systems analysis perspective on workflow and business results. A whole range of social conception is thus introduced in terms of location of people, conversations, informal relationships, etc. that better explains how responsibility and attention arise and are sustained in a group. (Indeed, the advantage is to understand *motivation* and *concern* as conceived and developed with respect to a group.)

To summarize, the key perspectives that we relate when comparing and promoting change in work practice are:

- *The physical work system* of people, technology, and organizations (including information-processing tools, facilities, and formal job specifications—things that can be directly changed),
- *The emergent activities* of the groups involved in doing the work (including especially how they conceive of their roles and knowledge when planning and handling trouble),
- *The workflow* (including both the physical transformations of work artifacts such as T.1 circuits and the transformations of representations such as forms describing circuits), and
- *The business results* measured by management for assessing customer satisfaction, competitiveness, and productivity.

In short, when someone suggests a change in work practice in the context of a work systems design project, they must have, at least implicitly, an argument about how the redesign will change or sustain business results over some period of time. These causal arguments are rarely articulated in detail, and are instead often truncated by assumptions about the downstream effects. For example, a designer might have preformulated a set of "good interactions in practice," assuming that if certain people-technology-organization interactions occur, the workflow and results will be acceptable or improved. Therefore, design conversations will focus only on how to bring about these interactions. But, as emphasized here, redesign requires some theory or experience that justifies the remainder of the argument. If a manager on the steering committee of a redesign project is focused on the number of days to process an order, he or she will probably only find the discussion of activities to be meaningful if the relation to workflow and business results is made explicit.

In general, ethnographic observation is required to identify situations that illustrate abstract design guidelines. For example, another change brought about in the back-end redesign is that pressure and rancor were replaced by "shared end-to-end responsibility." How are "pressure" and "rancor" manifest in activities? In the workflow? How is "shared end-to-end responsibility" manifest in activities and the workflow? To make such abstract, but central concepts explicit, anecdotes are converted into cases, whose preconditions and consequences are modeled, just as in traditional knowledge engineering. In effect, expressing such conventional social concepts in a formal model brings to social science a change in practice that complements the change in business process modeling in adopting the social perspective.

In summary, a design stance requires a *vocabulary* for describing practice (e.g., functional hand-off, alerting, discussing context, lack of contact, developed relationships, rancor, cryptic notes, responsibility) and a *theory* of how changes in jobs and technology will transform one work practice into another. This theory is necessarily quite detailed, for it must explain how the practice will develop (e.g., how do we get from "lack of contact between people" to "developed relationships"?) and *how this changed practice will affect workflow and hence business results* (e.g., how do "developed relationships" improve the quality of the work?). This means that we must organize ethnographic characterizations to portray shifts from "dysfunctional" to "desired" practice and make explicit the causal relations to the analysis provided by other system designers who are focusing on workflow, quality, and productivity. In practice, one might develop an initial model, knowing only that certain scenarios and behaviors are important, but without having an explicit theory of how they came about, what effects they are having on quality, and how to change them. Indeed, we usually build models to get such an understanding.

To conclude, Brahms is a tool for bringing a mixture of observational methods (ethnography) and insights about how work occurs to practical application in work systems design. As described, there are many other implications, ranging from instruction to the development of software agents. We believe the most important technical result from our work is the discovery that *a workflow diagram can be generated* from the local interactions between human agents, media (e.g., fax machine, computer terminal, phone) and representational artifacts (e.g., faxes, databases, voice mail). By building in fewer assumptions about information and workflow, essentially modeling how connections and reinterpretations occur in practice, Brahms models are potentially more useful for human-computer systems design than task and business process models alone. In particular, the idea of *knowledge variability* illustrates how synthesizing social and cognitive theories may be productive. Finally, the clarified distinction between tasks and activities provides a framework that changes how we view cognitive models, and suggests a broader foundation for studying and formalizing expertise.

APPENDIX: MODEL OF PRACTICE IN A HEALTH CARE MODULE

To further illustrate the value of activity-based modeling in Brahms, we sketch here a model from another domain of application. We developed a model of the work practice in a module consisting of about ten people in one corner of an outpatient clinic. The model describes the activities of people in this module, including their communications with other modules and the hospital. The *primary activity* we sought to model is the patient visit. Thus, the patient is the main character and the module is the stage. The climax scene is the physician's interaction with the patient in the exam room.

This particular module was chosen to test an electronic medical record (EMR) system for a large HMO. An ethnographic pre-study of several months was followed by a week of videotaping routine work in the outpatient setting. Our overall interest was in promoting participatory design of the EMR, plus understanding the difficulties and benefits of transitioning from a paper chart to an EMR.

We had available the following information for constructing this model:

- floor plan (blueprints)
- flow chart describing typical work process
- video tapes of 24 patient visits, each filmed at four locations having wide-angle and narrow views.
- ethnographers' field notes from several months of pre-study

- hour-long, transcribed interviews of members of the module, focusing on their experience with computers and potential uses of an EMR
- matrices comparing backgrounds and work styles of the providers
- sample forms and format description of a medical chart

Using Brahms, one might start by describing the groups (Figure A1).

The most important representational artifact at this work site is *the chart*. There are also two computer systems to be modeled, one for patient scheduling and other for patient data. The chart is modeled in terms of sections and what information is written in each section (e.g., Current progress note, new orders, lab and pathology results, back page). Other forms used by the caregivers and patients include: Pathology lab order, labels, and consent forms. Details about content and use of forms and computer systems are included in the model only as they arise in the description of activities. Other representational artifacts include messages from the appointment center in the clinic, a prescription refill (such as a fax from a pharmacy), phone messages from patients, and notes from physicians to nurses to call patients. Each of these is represented insofar as it is read, written, or hand-carried during a patient visit.

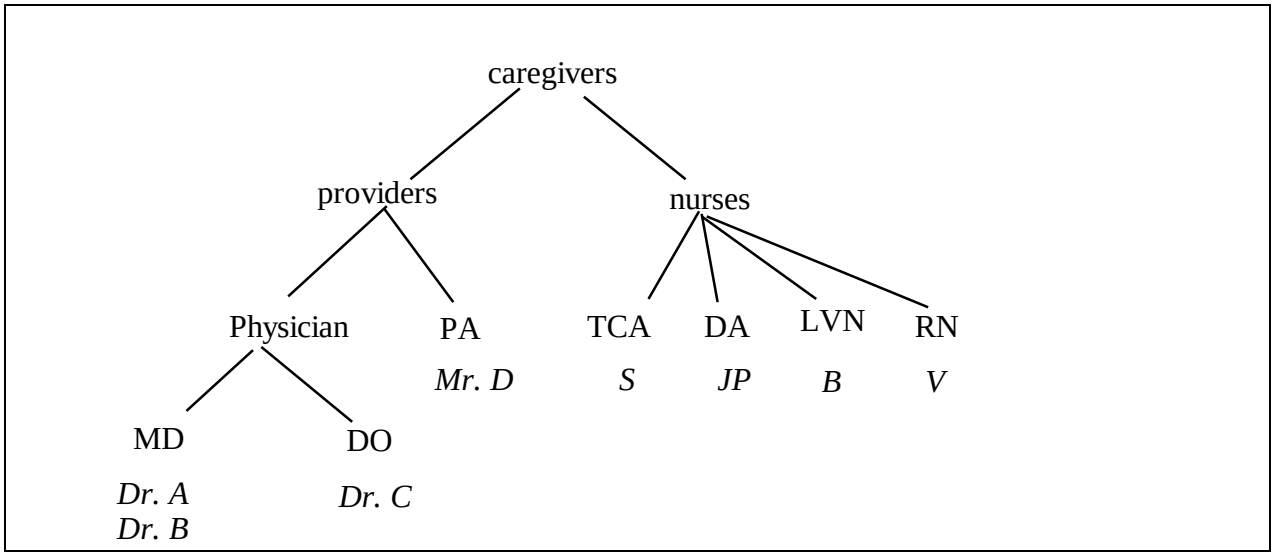


Figure A1. Groups for a health care module[6]
(Individuals who belong to each group are listed in italic.[7])

Early on, one describes the "geography" of activities, including the following locations:

[6]TCA = Trained Clinical Assistant, not a nurse technically.

[7] Omission of titles, e.g., "Ms" for nurses reflects status distinction in how nurses are viewed relative to providers.

- the clinic
    - the module
        - the waiting room
        - the nurse processing station (central nursing station)
        - the nursing station (4)
        - the exam room (4)
        - the provider's office (4)

Numbers in parentheses indicate how many instances of this kind of location appear in the module. Other locations that communicate with this location include the registration desk, the chart room, and the hospital.

The primary activity of the module is the patient visit. When not engaged in this activity, participants are engaged in the activity of "being in the module" and what they do may be omitted.

In scoping the model, we decided to distinguish the roles of the nurses and the capabilities of the providers (MD vs. physician assistant). We also wanted to model the *mentoring activity* between a physician (Dr. A) and a PA (Mr. D) (which occurs as a scheduled meeting).

The primary actors during the patient visit are: the patient, the registration clerk, the nurse (assigned to a provider for the day), the provider (MD or PA), and assisting nurse.

The main activities of the patient visit are, in order of occurrence (included optional activities are indicated by brackets):

- patient checks in for appointment (patient, registration clerk)
- patient arrives at the module (patient, nurse)
- initial interview by nurse (patient, nurse)

    —{pre-exam lab/pathology tests (patient, nurse)}
- examination by physician (patient, provider)

    —{drug administration (patient, nurse)}
- processing orders (patient, nurse)

    —{post-exam lab/pathology tests (patient, nurse)}
- patient checks out at registration (patient, registration clerk)

The *flow of people* is as follows:

- The Patient moves from registration desk to waiting room to nursing station to exam room and back out in reverse order.
- The Nurse moves from nurse processing station to her nursing station to her provider's exam room to her provider's office and then back to the central nursing station.
- The Provider moves from his office to his exam room and back to his office.

The *movement of the chart* for a patient visit is pivotal in the choreography of the visit:

- At the beginning of the day, the charts for a provider are delivered to the *nursing station* of the nurse assigned to that provider on that day.
- At the beginning of the day, each nurse moves the provider's charts to the *nurse processing station*.
- After the patient has arrived and been checked in, the nurse places the chart on the desk in the *provider's office*.
- The provider takes the chart with him into the *exam room*.
- After the patient exam, the provider takes the chart to his *office*.
- When the provider has completed his notes, he puts the chart in his *out-basket*.
- The nurse retrieves the charts periodically and takes them to the *nurse processing station*, placing them in the *central bin*.
- At the end of the day, the charts are picked up and returned to the *chart room*.

The "basic rhythm" of the work site is described chronologically (this is the formal schedule; the team interacts informally as needed):

- The clinic opens at 7:15 AM to 7:30 for incoming calls; chart room is open.
- Module reception (waiting room) opens 8:15 to 8:30 AM.
- Nurses arrive 8 to 8:15, officially 8:30, to get and organize charts, organize same-day appointments (SDA), schedule the PA, who arrives at 8:30.
- The MDs arrive at 9 and begin seeing patients at 9 AM.
- Patients are scheduled in half-day shifts, from 9 AM to noon and 1:30-4:30 (approx), every 10 minutes. There may be double appointments.
- If "doing triage in morning," MD works through lunch; afternoon appointments in module are first 1:30, last at 3 PM.

- Lunch begins for nurses at 12:15, but one covers until MDs are done.
- Each MD sees 10-14 patients during one half-day shift, 1 to 3 shifts per week.
- The PA sees patients 10 shifts per week. 7-8 shifts are SDA.

Note that providers have many other activities, including doing laboratory procedures, learning, mentoring at other locations, covering in the hospital ("team rounding"), evening emergency care, etc. This model only describes what providers do during a patient visit. That is, this is effectively a "model of the patient in the clinic" not a model of providers.

Similarly, we are aware of different ways nurses work together. For example, one nurse (RN or LVN) works with two providers on Tuesdays and Thursdays. Two or three other nurses (possibly including a TCA) work with three or four providers the other days of the week. The TCA prefers to work with Dr. A (the physician-in-charge), etc. Such interactional patterns illustrate part of what is meant by practice, as opposed to policy or procedures. We describe what people do together, how they interact with each other and objects in the environment.

In modeling the clinic, we focus on the nurses' first rule, "Keep the rooms filled." This orientations gives us the nurses' point of view, so we can expect what they might notice and helps us detect gaps in our model. This is an example of a choreography constraint, which describes what the group is trying to accomplish in their routine interactions. Accordingly, if a provider is absent on a given day, his exam rooms will be used for placing the third patient of another provider.

As an example, here is a sketch of the nurse's activity of coordinating patient visits, which occurs as a subactivity of "being a nurse in the module."[8] This sketch would be filled in by additional workframes that specify what the nurse does while waiting, etc. Or we could decide that because this model focuses on the patient, we will omit such details (corresponding to actors who leave the stage and return later).

[8]Key to notation: **NAMES OF WORKFRAMES IN ALLCAPS**; Key preconditions underlined (beliefs determined by detectables and/or thoughtframes); **Subactivities are composite actions,** others are primities (not decomposed); => indicates "go to location"; $X indicates variable binding to be referenced later; **Wait** indicates a detectable that causes an impasse until it changes state.

COORDINATING PATIENT VISITS

At the nurse processing station

If the light for provider $X appears on the console,

Subactivity:: The patient visit

THE PATIENT VISIT

=> Outside the waiting room

Retrieve the CPR from the slot

=> At the nurse processing station

Locate the chart and assemble materials (5 mins)

Take the chart

Subactivity:: Pre-exam patient interview

THE PRE-EXAM PATIENT INTERVIEW

=> In the waiting room

Call the patient

Tell the patient to follow you

=> In the nursing station

Stamp the progress note with the member's and provider's cards

Record the patient's vital signs on the progress note

Place the CPR and member's card on the front of the chart

Tell the patient to follow you

Note the room assigned to this visit ($X)

Subactivity:: Bring together patient and provider

Wait for the exam to be complete

When the provider indicates the exam is complete,

Subactivity:: Post-exam patient visit

BRING TOGETHER PATIENT AND PROVIDER

=> In the examination room ($X)

Tell the patient to undress and stay put

=> In the provider's office

Write the number of the examination room ($X) on the progress note

Place the chart on the provider's desk

POST-EXAM PATIENT VISIT

=> In the nurse processing station

If CPR indicates tests are ordered,

Prepare lab requisitions (3 mins)

Instruct the patient about the tests (2 mins)

Tell the patient to follow you.

Subactivity:: Direct the patient to the lab

Wait for the patient to return

If provider has requested a follow-up visit and date is less than 2 weeks,

Call the appointment center

Handle consultation requests

Tell the patient to return to the registration desk.

DIRECT THE PATIENT TO THE LAB

=> In the laboratory

Tell the patient to return to the nursing station when done.

To this point, we have only considered the routine work. Within this participation framework, we will want to represent what can go wrong, who detects problems where and when, and what outside influences affect the rhythm of activity in this work system. For example, the analysis so far suggests that we might focus on:

- Time spent processing (and finding) paper records at multiple locations.
- Difficulty interpreting x-rays that are kept at the hospital.
- Need to call pharmacy for dosage information (patient is left waiting in the exam room).
- Nurses having to locate lab results for providers.
- Effect of nurses having to cover other modules ("floaters"); affect of union rules on number of floaters.
- Different practices in other modules.
- How previous working relationships between these people affect current work practices.

This example illustrates the kind of analysis and model that might be useful for developing an electronic medical record. Rather than carving up the clinic into tasks performed by individuals, focusing on diagnosis and therapy, we first of all broaden the design space to consider information processing more systemically. A variety of different computer tools might be proposed, and their designs would be better related to the practices of the clinic.

ACKNOWLEDGMENTS

The original idea of a tool that would "make social processes visible" was developed at NYNEX S&T by a team led by Patricia Sachs in 1991-92, based on their experience in using SPARKS and development of the work systems design approach described in this paper. The Brahms architecture was conceived and developed by Bill Clancey (formerly at IRL) and Dave Torok (formerly at NYNEX S&T) in 1992-94. Our understanding of work systems design and social theories was developed through many long discussions with Gitti Jordan (IRL/Xerox-Parc), David Moore, Madeline Ritter, and Jim Euchner at NYNEX. Technical designs were contributed by Erik Vinkhuyzen, Alwin Bakkenes, and Edgar Steenvoorden. We thank Dan Russell, Peter Benda, Rich Keller, and the IJHCS reviewers for their many suggestions for improving this paper. We thank Ed Thomas (formerly vice president at NYNEX) for championing this work.

Brahms exists as a prototype developed in G2 (trademark Gensym Corporation) on a SUN workstation; a graphic interface implemented in Visual Basic displays simulation outcomes on a

PC. Current work includes comparative studies of tools and exploratory use on client projects. The name “Brahms” stands for “Business Redesign Agent-based Holistic Modeling System,” but it applies to any human activities.

The model of the health care module was developed with the cooperation and financial support of Southern California Kaiser Permanente. Judith Gregory provided the ethnographic data. Charlotte Linde (IRL) and Jean Gilbert (KP) were project managers. Support for the development of Brahms has been provided by NYNEX Science and Technology, Inc.

REFERENCES

Agre, P.E. (1997). Computation and human experience. Cambridge, UK: Cambridge University Press.

Brooks, R.A. (1991). Intelligence without representation. *Artificial Intelligence*. 47, pp. 139-159: Elsevier Science Publishers, B.V.

Brown, J. S., Collins, A., and Duguid, P. (1988). *Situated cognition and the culture of learning*. IRL Report No. 88-0008. Shorter version appears in *Educational Researcher*, **18**(1), February, 1989.

Burstein, M.H., Ferguson, W., and Abrett, G. (1993). A Simulation Development Tool for Evaluating Coordination Strategies in Organizations. *AAAI Workshop on AI & Theories of Groups & Organizations: Conceptual & Empirical Research,*. pp. 24-28.

Carley, D. Park, and M. Prietula. (1993). Agent honesty, Cooperation and Benevolence in an Artificial Organization. *AAAI Workshop on AI & Theories of Groups & Organizations: Conceptual & Empirical Research*. pp. 1-7.

Clancey, W. J. (1992). Model construction operators. *Artificial Intelligence*, 53(1): 1-124.

Clancey, W. J. (1993). A situated cognition perspective on learning on demand in *Proceedings of the Fifteenth Annual Conference of the Cognitive Science Society*. pp. 181-3. Boulder: Lawrence Erlbaum Associates.

Clancey, W. J. (1997a). The conceptual nature of knowledge, situations, and activity. In P. Feltovich, K. Ford and R. Hoffman (eds.), *Expertise in Context*, pps. 247-291. Cambridge, MA: The MIT Press.

Clancey, W.J. (1997b). *Situated Cognition; On Human Knowledge and Computer Representations*. Cambridge, UK: Cambridge university Press.

Clark, A. (1997). *Being There: Putting Brain, Body, and World Together Again.* Cambridge: The MIT Press.

Cohen, P. R., Greenberg, M. L., Hart, D. M., & Howe, A. E. (1989). Trial by fire: Understanding the design requirements for agents in complex environments. *AI Magazine* **10**(3), 34-48.

Corcoran, E. (1992). Building networks: New York Telephone rethinks how to regain lost customers. *Scientific American*, November, pp. 119-120.

Davenport, T.H. (1995). The fad that forgot people. *Fast Company.* November.

Dourish, P. Holmes, J., MacLean, A., Marqvardsen, P., Zbyslaw, A. (1996). Freeflow: Mediating between representation and action in workflow systems. *Proceedings of Computer Supported Collaborative Work*, pp. 190-8. Cambridge MA: ACM Press.

Ehn, P. (1989). *Work-oriented design of computer artifacts.* (2nd Edition). Hillsdale, NJ. Lawrence Erlbaum Associates.

Emery, F. E. (1959). *The Emergence of a New Paradign of Work.* Canberra, Australia: Australian Naional University, Centre for Continuing Education.

Emery, F. E., Trist, E.L.(1960). "Socio-Technical Systems." In C.W. Churchman and others (eds.), *Management Sciences, Models and Techniques*. London: Pergamon.

Fafchamps, D. (1991). Ethnographic workflow analysis: specifications for design. In Human Aspects in Computing, Design, and Use of Interactive Systems and Work with Terminals. *Proceedings of the Fourth International Conference on Human-Computer Interaction* (Amsterdam, 1991), H.-J.Bullinger, Ed., Elsevier, pp. 709-15.

Gilbert, N. and Doran, J. (1993). *Simulating Societies: The computer simulation of social phenomena.* UCL Press.

Greenbaum, J., Kyng, M. (Eds). (1991). *Design at work: Cooperative design of computer systems*. Hillsdale, NJ. Lawrence Erlbaum Associates.

Gustavsson, R. (1993). Societies of Computation: A framework. *AAAI '93 Workshop on AI & Theories of Groups & Organizations: Conceptual & Empirical Research.* pp. 96-102.

Hayes-Roth, B., Brownston, L., and Sincoff, E. (1995). Directed improvisation by computer characters. *Proceedings of the International Joint Conference on Artificial Intelligence.*

Hayes-Roth, F., Waterman, D., and Lenat, D. (eds.). (1983). *Building Expert Systems.* New York: Addison-Wesley.

Hutchins, E. (1995). *Cognition in the Wild.* Cambridge: MIT Press.

Joosten, S.M.M. and S. Brinkkemper. (1996). Fundamental Concepts for Workflow Automation in Practice. In S. Wrycza and J. Zupancic (Eds.), *Proceedings of the Fifth International*

Conference Information Systems Development (ISD'96): Methods and Tools, Theory and Practice. pp. 311-322, ISBN 83-86230-18-5. Gdansk, Poland.

Levitt, R. E., Jin, Y., Oralkan, G.A., Kunz, J.C., and Christiansen, T.R. (1995). *Computational enterprise models: Toward analysis tools for designing organizations.* CIFE Working Paper, Stanford University, Department of Civil Engineering. February.

Malone, T.W. and Crowston, K. (1991). *Toward an interdisciplinary theory of coordination.* Technical Report CCS TR# 120, Center for Coordination Science, MIT.

Menzies, T., (in press). Is knowledge maintenance an adequate response to the challenge of situated cognition for symbolic knowledge-bases systems? *IJHCS*, Special issue on Situated Cognition. pps. ???

Mills, C. W. (1940). Situated actions and vocabularies of motive. *American Sociological Review*, 5: 904-913.

Mitchell, M. (1993). *Analogy-Making as Perception: A Computer Model.* Cambridge: Bradford Books.

Nakashima, H., Noda, I., and Handa, K. (1996). Organic programming language GAEA for multi-agents. *Proceedings of the Second International Conference on Multi-Agent Systems.* December 10-13, 1996, Kyoto, Japan. pps 236-243. Menlo Park: AAAI Press.

Nagel, T. (1986). *The View from Nowhere.* New York: Oxford University Press.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Nonaka, I., H. Takeuchi. (1995). *The Knowledge-Creating Company*. New York: Oxford University Press.

Orr, J. E. (1995). Ethnography and organizational learning: In pursuit of learning at work. In S. Bagnara, C. Zuccermaglio and S. Stucky (eds.), *Organizational Learning and Technological Change*, pp. 47-60. Berlin: Springer.

Pasmore, W.A. (1994) *Creating Strategic Change: Designing the Fliexible, High Performing Organization.* New York: Wiley.

Resnick, L. B., Levine, J. M., and Teasley, S. D. (editors). (1991). *Perspectives on Socially Shared Cognition.* Washington, D.C.: American Psychological Association.

Sachs, P. (1993) "Shadows in the Soup: Conceptions of Work and the Nature of Evidence" in *Newsletter of the Laboratory of Comparative Human Cognition*, Fall ISSUE

Sachs, P. (1995). Transforming Work: Collaboration, Learning, and Design. *Communications of the ACM*, Vol. 38, No.9, pp.36-44.

Salgame, R. R., Becker, S.G., and Yu, D. H. (19??). SPARKS: A knowledge-based process modeling and simulation system. Proceedings??? Pages??? Publisher???

Schön, D.A. (1983). *The Reflective Practitioner; How Professionals Think in Action.* Basic Books, Inc.

Steels, L. and Brooks, R. (Eds.). (1995). *The "Artificial Life" Route to "Artificial Intelligence": Building Situated Embodied Agents*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Tambe, M, Johnson, W.L., Jones, R.M., Laird, J.E., Rosenbloom, P.S., and Schwamb, K. (1995). Intelligent Agents for Interactive Simulation Environments. *AI Magazine*, **16**(1):15-39,Spring.

Tokoro, M. (Ed.) (1996). *Proceedings of the Second International Conference on Multi-Agent Systems.* December 10-13, 1996, Kyoto, Japan. Menlo Park: AAAI Press.

Tyo, J. (1995). Simulation modeling tools. *Information Week*, 60-67. July 10, 1995.

Weickert, K., (1995). *Design under Uncertainty: Engaging the Context of Use in the Design of Expert Systems.* Dissertation in Information and Computer Science, University of California, Irvine.

Weisbord, M.R. (1987).*Productive Work Places: Organizing, and managing for dignity, meaning, and community*. Jossey-Bass Inc, Publishers,CA.

Wenger, E. (1997) *Communities of practice; Learning, meaning, and identity*. New York: Cambridge University Press.

Whalen, J. and Vinkhuyzen, E. (forthcoming). Expert systems in (inter)action: Diagnosing document machine problems over the telephone. In Christian Heath, John Hindmarsh, and Paul Luff (eds.), *Workplace Studies: Recovering Work Practice and Informing Systems Design.* New York: Cambridge University Press.

Wynn, E. (1991). "Taking practice seriously." In J. Greenbaum and M. Kyng (eds.), *Design at work: Cooperative design of computer systems*, pp. 45-64. Hillsdale, NJ: Lawrence Erlbaum Associates.

Zilberstein, S. (1996). Resource-bounded reasoning in intelligent systems. *ACM Computing Surveys*, Vol. 28, No. 4 (Dec. 1996), Pages 15-16.

Zuboff, S. 1987. *In the Age of the Smart Machine: The Future of Work and Power* . New York: Basic Books.