The Adoption and Use of 'Babble': A Field Study of Chat in the Workplace*

Erin Bradner

University of California at Irvine, USA

ebradner@ics.uci.edu

Wendy A. Kellogg, Thomas Erickson

IBM T.J. Watson Research Center, USA

wkellogg@us.ibm.com, snowfall@acm.org

Abstract. One way to gain a principled understanding of computer-mediated communication (CMC) use in the wild is to consider the properties of the communication medium, the usage practices, and the social context in which practices are situated. We describe the adoption and use of a novel, chat-like system called Babble. Drawing on interviews and conversation logs from a 6-month field study of six different groups at IBM Corporation (USA), we examine the ways in which the technical properties of the system enable particular types of communicative practices such as waylaying and unobtrusive broadcast. We then consider how these practices influence (positively or negatively) the adoption trajectories of the six deployments.

Introduction

The CSCW and related literatures are well populated with empirical studies of groupware adoption (e.g., Ehrlich, 1987; Grudin & Palen, 1995; Orlikowski, 1992). In particular, social factors impacting the adoption and use of computer-mediated communication (CMC) technologies in the workplace have been studied for over a decade (e.g., Ehrlich 1987; Markus & Connolly, 1990; Sproull & Kiesler, 1991). These studies reveal ways in which behavior and social conventions affect adoption, a common conclusion being that understanding adoption requires careful examination of the interactions between technological features and the social context of use. As Ehrlich (1987) put it: "planning for successful [CMC] adoption requires knowledge of individual and organizational communication patterns and the relationship between those patterns and particular communication systems solutions." Yet specific examples of communication patterns and the interactions between patterns and features of CMC technology are largely absent from the literature, as are conceptual tools for detecting and describing such interactions. One contribution of this paper is to address this absence by closely examining the how features of a newly deployed system supported particular communicative practices and thus stimulated, or suppressed, its adoption. A second contribution is to investigate the use of chat in the workplace. While many studies examine chat (e.g., MUDs and MOOs) in academic and social settings (e.g., Correll 1995; O'Day *et al.,* 1996; Turkle, 1995) and email in organizations (e.g., Sproull & Kiesler 1991), few look at the use of chat in business settings (see Churchill & Bly, 1999; Kovalainen *et al.,* 1998).

In this paper we present a field study of the adoption and use of a system called "Babble" by six groups in a large U.S. corporation. We begin with background: the deployment process, field study, and system interface. Next, we examine how communicative practices arise from interactions among the technical properties of the system and the social characteristics of the groups, and how they stimulated or suppressed adoption. Finally we discuss critical mass, social affordances and interaction ecologies, concepts which, we believe, can provide a basis for a better understanding of adoption.

The Babble Deployment and Field Studies

Site Selection and Deployment

Six groups at the IBM T. J. Watson Research Center were studied, including a Software Engineering, Staff, and Human Resources group, a professional and a social cohort, and "the Babble Lab" group (which developed the system). Group size ranged from 5 to 175 people. The Software Engineering, Staff, and Human Resources groups were collocated (i.e., had adjacent offices) and organizationally bound (i.e., members belonged to the same department). Cohorts were geographically distributed but shared professional or social interests. The Babble lab's members were co-located and organizationally bound. (See Table I).

Groups were recruited via in-house demonstrations. Each received a separate discussion server. The client software included on-line help, but barring rare interactions between the authors and individual users, no formal training or usage guidelines were

provided. Babble use was entirely discretionary.

Field Work and Data Collection

Prior to deployment, the first author observed the physical distribution of offices and conducted unstructured interviews investigating group work practices (e.g., delegation of work) and culture (e.g., perceptions of information sharing). After deployment, additional interviews were conducted at approximately four-week intervals. Throughout deployment, at frequent, unscheduled times, live Babble discussions were observed. Most observations and interviews were conducted over approximately a twelve-week period during the summer of 1998, with additional interviews in December, 1998. The Babble system automatically archived all conversations (with the exception of private chats).

Table I summarizes the deployments. 'Adoption status' reflects the subjective opinion of the workgroup members themselves. 'Use' shows the number of months of logs for each group.

Babble Deployment Summary					
Group Name	**Group Purpose**	**N**	**Local:Remote**	**Use (mos.)**	**Adoption Status**
Babble Lab	Software Design	12	10:2	18	Adoption
Software Engineering	Software Engineering	6	6:0	6	Adoption
Human Resources	Personnel Mgmt.	10	8:2 (adjacent offices)	3	Non-adoption
Staff	Staff	5	5:0	2	Non-adoption
Social Cohort	Summer Interns	175 12*	Non-collocated	3	Adoption
Professional (HCI) Cohort	HCI Researchers	28 8*	Non-collocated	3	Adoption

* Indicates the number of cohort members who actually used the system.

Table I: Deployment Summary

The Babble Environment

In this section we describe the interface and functionality of the Babble environment, and provide an example of use by one group of long term users.

The Babble Interface

Babble is a chat-like communication tool in which typed messages are transmitted across a TCP/IP network, stored on a server and displayed to each client. Babble allows its users to engage in synchronous or asynchrovnous textual conversations, and provides visual feedback regarding who has recently participated in a conversation (see Erickson *et al.*, 1999).

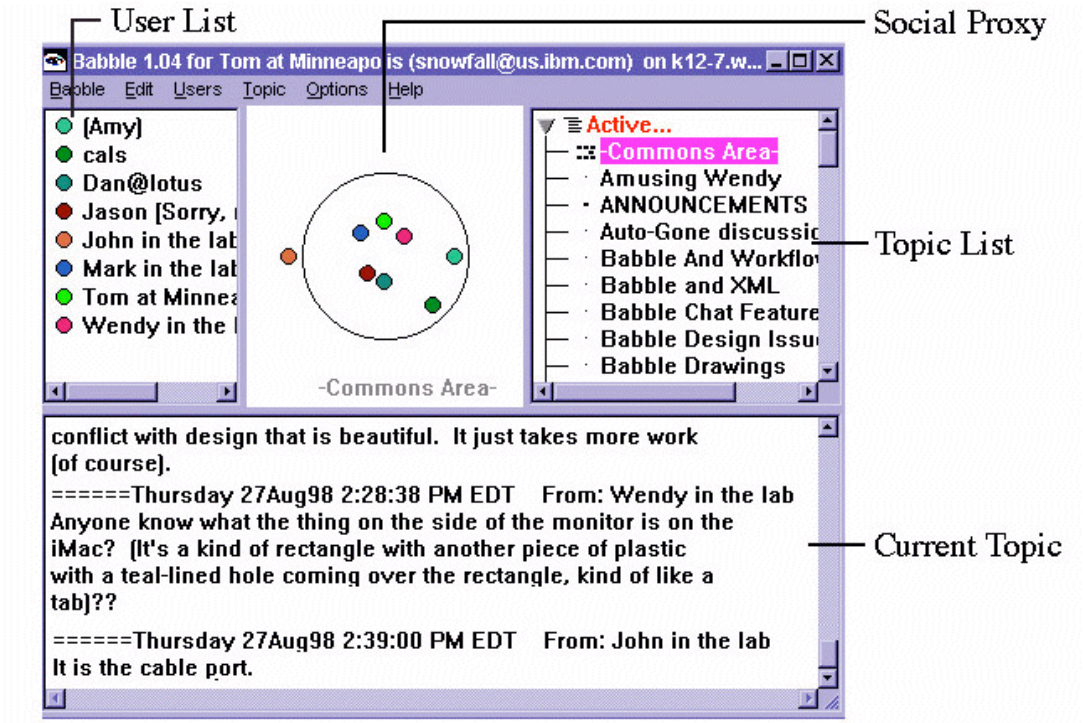

Figure 1: The Babble Interface

The panes of the Babble window (Figure 1) display the following information: a list of all connected users; the social proxy (a minimalist graphical representation of user activity); a list of topics (user-defined conversation areas); the current topic (i.e., text of the conversation). Messages appear in the order posted.

Three features of Babble distinguish it from other chat systems. First, Babble conversations are persistent: the conversations stay on the server permanently, thus permitting asynchronous conversations and activities. A user who is not on-line when a comment is made can see it later, and can scroll back through the entire history of a conversation. Second, a minimalist graphical representation called a social proxy is used to provide information about who is currently present in the conversation. The proxy uses a large circle to represent the conversation, and colored dots (a.k.a. "marbles") to represent individuals. A marble inside the circle represents a user who is 'in' the displayed conversation; a marble outside the circle is in some other conversation. When a user interacts with Babble - either by posting a message, or simply by scrolling or clicking on the interface - her marble rapidly moves towards the center of the circle; with inactivity the marble will slowly drift out to the inner edge of the circle. In Figure 1, five participants have recently 'spoken' or 'listened,' two have been idle, and one is in a different conversation. The third distinguishing feature of Babble is that it lacks technical mechanisms for enforcing behavior. Originally intended for small workgroups, it provides no technical means for 'kicking' people off, creating private topics, etc. With the exception of private, one-to-one chats, all Babble conversations were visible to everyone in a deployment group. Although various usage conventions have arisen, all negotiation and enforcement of such conventions is social.

An Example of Babble Usage

As the longest running Babble, the Lab Babble (Table I) provides a glimpse of how a successfully adopted Babble might evolve. It has grown steadily as measured by the amount of information and number of topics it houses, and the number and variety of communicative practices its inhabitants use.

Overall, the Lab Babble can be characterized as a core of synchronous activity surrounded by a constellation of asynchronous conversations. At the center of activity is the Commons Area, a place where collocated and remote members share news, engage in banter, get help, and 'hang out.' Analysis of Commons Area conversation over two-weeks showed a more synchronous temporal rhythm than other topics, and a mix of work and social talk. Approximately 50% of messages are social, including greetings, chit-chat, general announcements, and non-work questions; over half (65%) of these were posted by remote users (at home, traveling, etc.). Other topics tend to be used asynchronously (hours or days often separate comments) and largely, though not exclusively, for work-oriented discussion or information sharing. Of fifty-nine other topics: 8 are non-work related (e.g., Dilbert Trivia, Bad Jokes), 8 are group maintenance topics (e.g., Announcements), and the remaining 43 are project or topic related.

Communicative Practices in Babble

In this section we look at four communicative practices that emerged among the six Babble deployments. We describe each practice, and discuss how it arises from the technical properties of the Babble system and the social properties of the user groups. In the subsequent section we describe the influence that these communicative practices had on the success or failure of the adoption of Babble.

Waylay

Babble enables its users to lie in wait for, or "waylay" their coworkers. In terms of simple behaviors, it is possible to log into Babble, wait for signs that a particular person is around, and then open a communication channel with that person. More specifically, five properties of the Babble system enable waylaying:

1. Babble displays a list of participants (and updates when newcomers arrive)
2. Babble identifies every participant by name
3. Babble allows the name of the topic each user is reading to be displayed
4. Babble maintains an open communication channel among users once a user has logged in or once a private chat is initiated
5. Babble shows visually whether a person has recently read or written to a topic or has recently changed their state of activity.

Properties 1, 2 and 3 enable a user to determine the arrival, identity and current location of a user. Property 5 shows when a user is actively attending to the conversation topic in which she is located. In particular, at the moment the person being sought logs on or interacts with the interface after being idle, her 'marble' moves abruptly to the center of the circle, a movement that easily attracts attention. Thus, property 5 shows that the person being sought is attending and able to see a request for attention from the person doing the seeking. Property 4 allows the user to engage the attention of another user by addressing a comment directly to that user within the conversation in which she is currently active, or by opening a private chat window onto her screen. In addition, the person being sought becomes socially accountable to the person doing the seeking, because the person being sought *knows that the person seeking her knows* that she is paying attention to Babble (property 5), and has seen the request (property 4).

Just because Babble makes waylay technically possible is not proof that it occurs. However, our data show that waylay does occur, and that it differs depending on context. Two types of waylay are evident in our interviews. One type can be construed as an act of expertise selection (McDonald & Ackerman, 1998); for example, users might open a private chat window on an expert's screen and leave it open until the expert responds. This type of waylay is similar to 'camping on' a telephone extension or video connection (Fish *et al.,* 1993). A summer intern explains how he used waylay to get answers to programming questions:

> "Most of the people who are on now aren't programmers so I barely ever ask [programming questions]. When Eugene comes on, I ask him because he's a Java

> programmer." - Intern

Another type of waylay occurs when a manager uses Babble to assign work to a subordinate. Managers often walk a fine line between what they see as necessary socializing (such as inquiring how someone is 'doing' before making a work request) and the desire to be productive. A manager explains this tradeoff:

> "When I go to [my assistant], because you have to have a proper human interchange when you do that, usually I have to start with a little chat firstÖ 'how is everything? Good.' And then you get to: 'Could you do this for me?' But Babble has the potential for cutting off some of the extraneous noise that maybe you don't want. So there is a balance between 'how's everything going?' and getting on to the next task. - Human Resources Manager

In addition to the technical properties of the Babble interface that enable waylay, the existence of social relationships between user dyads (information seeker and expert, manager and subordinate) make waylay a socially viable interaction.

Unobtrusive Broadcast

Another type of communicative practice is what we call "unobtrusive broadcast" of information. Unobtrusive broadcast stands in marked contrast to waylay; users reported that they often used Babble over other forms of communication because it enabled them to request or share information *without* interrupting others.

An example of a question and answer exchange from the Software Engineering group (Figure 2) illustrates unobtrusive broadcast.

==Thursday 6Aug98 11:40AM From: Peter

Is there a problem with marvin? I am trying to run textract on Dhaka and even though I have exported DICTPATH I am getting a LEX: No POE DICTPATH error.

==Thursday 6Aug98 11:41AM From: Megan

Not that I know of. Perhaps it's your /tmp directory that is too full. Try emptying it

==Thursday 6Aug98 11:57AM From: Peter

Responding to: <<Perhaps it's you /tmp directory that is too full>>

Yes, that was it. Thanks...

==Thursday 6Aug98 12:11PM From: Megan

It's the beauty of BAI - it fills the /tmp directory with garbage and eventually it can't do so anymore. No msg, no warning, it took Karen forever to figure this one out!

Figure 2: Unobtrusive broadcast in the Software Engineering Group

This excerpt shows a question broadcast to the Software Engineering group. In it, Peter, a relatively new member, posts a question in the Commons Area regarding an error message he encountered while using an unfamiliar system. Shortly after his post, Megan responds with a solution (later attributing it to Karen). In follow up interviews, these participants explained that the origin of the problem was obscure and the solution was extremely non-intuitive. Peter explained that Babble allowed him to get the help he needed while not overburdening his more senior colleagues.

There are a number of technical properties of the Babble system that support unobtrusive broadcast:

1. Babble maintains an open communication channel among users
2. Babble is persistent (e.g., it stores messages indefinitely)
3. Babble concatenates and displays a topic's messages within a single window
4. All messages are displayed in the order in which they are generated.

Property 1 enables synchronous, distance communication among users. Property 2 enables asynchronous communication such that messages (e.g., questions) posted when a user is absent, or not attending, can be read and responded to at any time (presumably when it is convenient for the respondent to do so). Property 3 enables messages to be broadcast and easily read – since all a topic's messages are in a single window, users do not open separate windows to read each message. Property 4 guarantees that answers appear after questions, so that all users can easily tell that a question has been answered simply by scanning the display. This in turn helps avoid duplicating answers and permits users to build upon each other's work, for example, elaborating a previous answer or suggesting alternatives (cf. Ackerman & Palen, 1996).

Users appreciated the unobtrusiveness supported by these properties. When asked how Babble was used, a software engineer responded:

> "[Babble] is used to ask a question of everybody if somebody needs something, not even very urgently, but it is a very unobtrusive way of asking everyone for help. It is very fast and gives immediate response [from other people] without being intrusive." - Software Engineer

With regard to choosing Babble over a face-to-face interaction with a specific colleague, another software engineer commented:

> "Karen doesn't like to be disturbed in her office, when you knock on her door and ask her a question. She does not get disturbed at all when you post it on Babble." - Software Engineer

With regard to choosing Babble over email, a staffer explained:

> "When you have a little question and you send it out [using email] to four different people and then over a period of time you get four different answers back, you are wasting three people's time there. If you try to let them know that somebody already answered you, you are making more unnecessary email. I would rather use Babble." - Staffer

These, and other comments from interviews, suggest that Babble was well liked because it minimized disruption to colleagues when information was being sought, and avoided "wasting people's time." Users also felt that Babble lessened the perceived assertiveness of question asking. The alternatives – yelling down the hall, knocking on doors, or sending broadcast email, were viewed as inappropriate, potentially disturbing, and a nuisance, respectively.

Staying 'In the Loop'

When Babble is used for questions and answers, not only do all users know a question has been answered, they also know who asked the question, who responded, and what the answer was. To the extent that Babble users share the same work context, such exchanges also provide a basis for inferring how a project is progressing, who is working on what, and so on. Thus, a side effect of public information broadcast is passive information sharing and an increased awareness of the activity of others that is idiomatically called being 'in the loop.'

The same properties of Babble that support unobtrusive broadcast also support activities such as 'talking out loud' and peripheral monitoring. An excerpt of conversation among members of the Software Engineering team (Figure 3) provides examples of 'talking out loud' (lines 1–6), monitoring by the group manager (lines 8–10), and 'overhearing' by a coworker (lines 13–15). The excerpt reveals how the Babble system's time-stamping and speaker identification support the practice of peripheral monitoring. Reading the text in Figure 3, we learn what Paul has been working on for two days (a software release) and that he had problems with a file (the op_tools file). From these posts the manager, George, concludes that Paul is making progress and asks him to provide a solution to a looming problem (how to build RTPS), which Paul does (use release numbers to track changes). Approximately two hours later, Karen 'overhears' the conversation and objects. Because discussion between Paul and George is available to Karen, she is informed of and, more to the point, involved in a decision that concerns her.

---

1. ======**Monday 31Aug98 4:04:14 AM EDT From: Paul in France**
2. Friday afternoon, before i left work, i completed a build of the system....
3. ======**Monday 31Aug98 8:35:57 AM EDT From: Paul in France**
4. Peter is the person assigned here in Lyon to study Silvercom and see what they have...
5. ======**Monday 31Aug98 11:28:19 AM EDT From: Paul in France**
6. i've built op_tools several times now with different options and after fooling with...
7. ======**Monday 31Aug98 11:37:05 AM EDT From: George**
8. Paul,
9. It sounds as if you're making progress. One thing I've not heard much about yet (perhaps it's in the part of Babble I haven't read yet) is details about how we're going to use RTPS. What mechanisms are you building to make it "as easy as bbvc" for us?
10. If I've missed discussions of this stuff, please give me pointers.
11. ======**Monday 31Aug98 11:44:09 AM EDT From: Paul in Lyon**
12. at this point, separation will have to be done by different release numbers for different changes, and it is up to us to negotiate which things appear in the next release. anything not specifically accepted for the next release of the product has to be done as part of a "research-only" release and integrated back in manually if that status changes later.
13. ======**Monday 31Aug98 1:56:38 PM EDT From: Karen**
14. Responding to: <<integrated back in manually>>
15. Excuse me, but I somehow thought manual integration and acute versionitis were supposed to be only one-time occurrences, after the initial switch-over.

Figure 3: Examples of Talking Out Loud and Peripheral Monitoring

These sorts of practices provide a way for group members to share information, both synchronously and asynchronously, while

These sorts of practices provide a way for group members to share information, both synchronously and asynchronously, while placing minimal demands on coworkers. Furthermore, 'talking out loud' frees the speaker of the burden of deciding who needs to be kept informed.

> "I think the main advantage to Babble as opposed to voice mail or email is being able to keep track of a whole set of discussions so other people also know what is going on. One of the best features Ö is being able to keep other people abreast of what is going on without sending email. Or without having to call. Or getting up out of my chair [laugh]." - Software Engineer

Interview data show that the desire to 'talk out loud' about projects and to peripherally monitor others' work activity motivated Babble use in all groups. These behaviors help users stay apprised of news, the status of each other's projects and changes in the work environment. Because Babble also supports unobtrusive broadcast, it makes 'being kept in the loop' less onerous.

Our analysis of peripheral monitoring in Babble recalls several studies of information flow that reveal communicative practices used to synchronize activities and integrate information in the workplace (e.g., Heath & Luff, 1991; Watts *et al.,* 1996). These studies focus on auditory and gestural communication surrounding event-driven collaboration, such as real-time coordination of London's underground trains and space shuttle mission control. Although Babble communication is less temporally interdependent and more asynchronous than in these event-driven contexts, we also observed a dual role of workplace communication. Like train controllers who speak out loud in the control room and mission controllers who use open voice channels, Babble users broadcast messages both to communicate their activities and to allow other group members to observe the consequences of their own actions (see Watts *et al*., 1996).

Discussion Sanctuary

Babble provides secure communication and access control. A firewall restricts access to IBM users, and separate servers for each group ensure that users access only their own Babble. Server access requires only the appropriate client, so a group regulates access by controlling distribution of its client.

Interview data show that the technical properties of secure communication and access control promoted the feeling that Babble was a 'safe' place to talk. Informants stressed that restricting access to 'members only' promoted informal conversation, a free-flowing exchange of ideas, and social banter.

> "I think [Babble is] less formal. I treat it less formal. I wouldn't write mail about someone else's bug unless I check very very carefully that it is indeed in their code. ... It's funny but it's OK to write things [in Babble] that are not 100% finished... not that thought through...half-baked ideas are OK. Somehow it's much more like conversation." - Software Engineer

Users also said that they were less careful about the mechanical aspects of writing using Babble (e.g., as compared to email) because they knew that Babble discussion was confined to the group. For example:

> "When you are in Babble it seems like a more relaxed atmosphere and you don't have to watch your spelling, you don't have to have your sentence structure perfect and all that. [In email] you feel like everything has to be correct." - Recruiter

That Babble provides a safe sanctuary becomes quite evident when members perceive that quality being threatened. This occurred in the Software Engineering group when a client joined Babble. Several core members voiced strong concerns that the client's presence threatened the integrity of their Babble, for example:

> "Peter asked the person in Lyon to be on all the time. So I think to myself, 'is she listening to every word?' Once you start being very careful [about what you say] then you start to lose something essential to the discussion." - Software Engineer

These concerns are serious because Babble's access control is all or none: once 'outsiders' are allowed in, they can see everything. Conversely, because Babble makes user actions visible via the social proxy, and provides ways of finding out who has been in a topic, 'outsider' behavior can be monitored.

To summarize, we have examined four communicative practices: waylay, unobtrusive broadcast, staying 'in the loop' (including 'talking out loud' and peripheral monitoring), and discussion sanctuary. These practices emerge from the technical features of the Babble environment and the social dynamics and practices of each workgroup. They provide grounding for understanding the patterns of adoption and non-adoption that we observed, to which we now turn.

The Adoption of Babble

In this section we look at the adoption (or lack thereof) of the various Babble deployments. We analyze adoption in terms of the communicative practices Babble supports, and the ways in which those communicative practices support or conflict with the social dynamics and work practices of the groups.

The First Month

Initial use does not necessarily mean that a system will be adopted: all six Babbles had significant use in the first four weeks (Figures 4 and 5), especially in total postings (Figure 6). Only after 4-6 weeks is there any evidence of adoption failure

(Figures 4 and 5), especially in total postings (Figure 6). Only after 4–6 weeks is there any evidence of adoption failure.

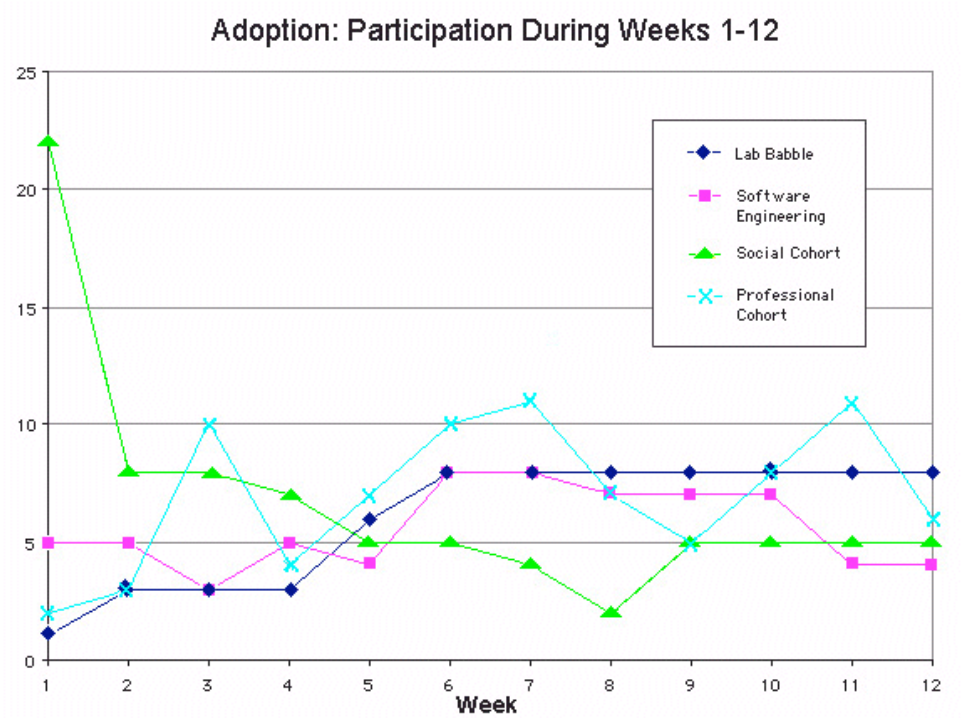

Figure 4: Number of Users Who Posted in Babble (Adopters)

Logs showed that Babble was used for both private and group conversations. We put conversation into four categories: interactive discussion, announcements, requests for information, and dedicated topics. Interactive discussion included greetings, shop talk, and social chit-chat; it was subdivided into social and work-related discussion. Announcements covered meetings, out-of-office messages, and workplace news (social or work related). Requests for information included technical assistance, business-related and social inquiries (and responses). Dedicated Topics included asynchronous interaction associated with a project or topic area (Adams *et al.,* 1999). Figure 6 summarizes the first four weeks of use.

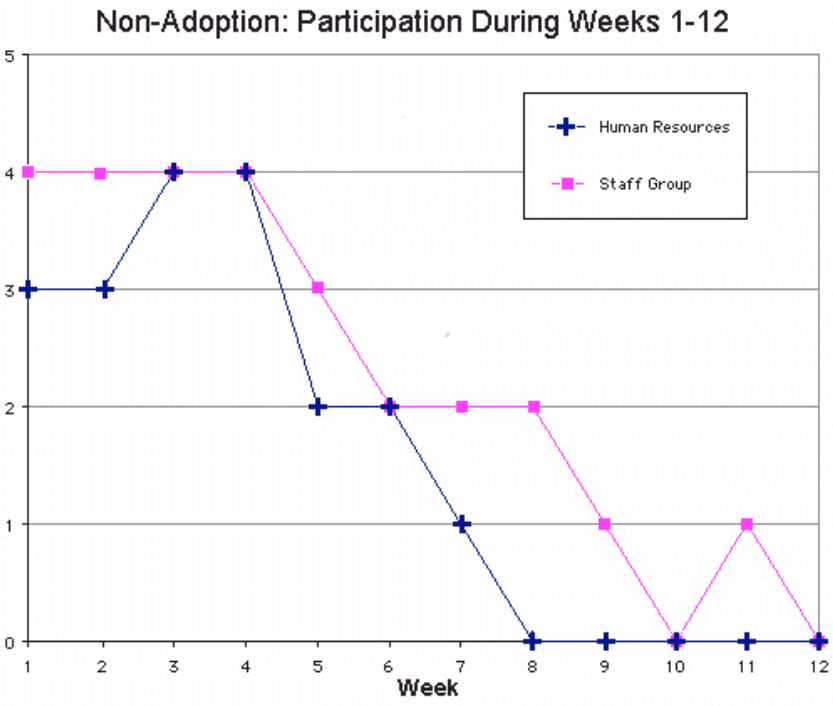

Figure 5: Number of Users Who Posted in Babble (Non-Adopters)

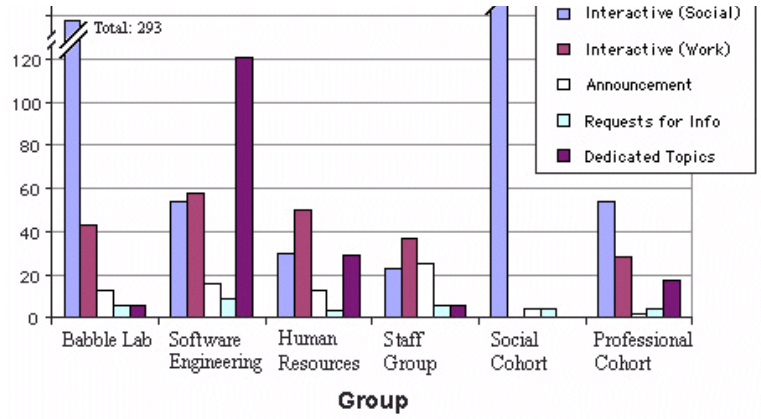

Figure 6: Total posts in the first 30 days of use, broken down by category.

Although one might expect groups using the same software within the same organization to exhibit somewhat similar behavior during the first month of use, Figure 6 shows varied usage patterns among the six groups. Interactive social discussion dominates in three groups, whereas use of dedicated topics dominates the Software Engineering group. The Human Resources and Staff groups show a slight dominance of interactive work discussion with relatively little social discussion. In short, although participation is relatively high for all groups during the first month (Figures 4, 5), the content varies considerably during this time (Figure 6), indicating that differences among groups in social dynamics and work practices affect the earliest stages of use. This suggests that groups appropriate new technology differently; understanding the relationship between group dynamics and adoption is the topic of the next section.

The Communicative Practices and Adoption Relationship is Complex

How might we understand an initial burst of usage followed by failure (or not) of adoption? Our data suggest that one answer lies in looking closely at the communicative practices that Babble supports. Communicative practices affect adoption by making certain social behaviors more pronounced. If Babble enables a practice that is not desired by some group members, or that produces consequences some members dislike, adoption may fail.

Waylay is a good illustration of the relationship between communicative practices and adoption. As noted elsewhere (Ackerman & Palen, 1996), electronic waylay for the purposes of soliciting technical assistance appears to reinforce ongoing use of chat as a distributed help environment. Our data bear this out: high levels of work-related interaction (Figure 6) can be associated with successful adoption in the Software Engineering group (Figure 4).

However, when waylay is or can potentially be used to assign work, it inhibits use. Interview data strongly indicate that the fear of waylay was a strong deterrent to use among the members of one group. After an initial period of several weeks during which most of the group was using Babble, the group failed to adopt. One member of this group reported that he and his colleagues were hesitant to log in to Babble because it made them too accessible. He explained that he and his coworkers feared that their increased accessibility might enable their manager to rope them into additional projects. In effect, the increased accessibility created by participation in Babble threatened their autonomy. Our informant described the fear of waylay below:

> "I'm [near] Mike [the manager] and Susan isn't that far away. When random [unassigned] work comes in, it sticks to us. Babble means that everyone is immediately accessible, so people who aren't geographically near Mike [the manager] are thinking that this accessibility is a down side [to Babble]." - Staffer

Although the use of chat to waylay subordinates for the purposes of assigning work is a blessing for managers because it reduces the social overhead required to engage a subordinate (discussed above), it may be considered a curse by workers. Thus, the communicative practice of waylay reinforces or inhibits adoption among different sub-groups within an organization.

Similarly, the use of Babble to maintain a closed discussion sanctuary encouraged use in the case of the Babble Lab and Software Engineering group (Figure 6). Babble encouraged free exchange of ideas, uninhibited brainstorming and casual chit-chat in these groups. At the same time, by using Babble to create and protect a discussion sanctuary, groups could hobble growth of participation. For example, usage logs and interviews indicate a "cliquish" behavior among the social cohort (i.e., the summer interns). In this deployment, access to Babble was extended to the entire population of summer interns. However, within a week of deployment a small faction of eight people established a sanctuary for their 'Babble club' excluding all others. The initial spike followed by flat line at five users in the usage graph (Figure 4) bears this out. In an interview, an intern revealed that uninvited users were deliberately alienated through vituperative comments:

> "...a couple of the other people decided that they were going to have a Babble "club" for the cool Babble people. So people kept [deliberately] getting into arguments the first couple of days about all different things. They were just trying to get rid of some people." - Intern

In summary, our examination of the differential adoption of Babble has revealed a number of issues. First, early usage, even if it is fairly extensive, does not necessarily mean a system will be adopted. Indeed, our data suggest that at least in some cases, initial usage leads to the evolution of communicative practices which, as they play out in the work situation, may have consequences that

usage leads to the evolution of communicative practices which, as they play out in the work situation, may have consequences that are not be desired by the group, thus leading to rejection. Furthermore, even when a system is adopted, it is important to ask by whom it is adopted and to what ends: adoption of a CMC system by a group, or portion of a group, is a more gradual and complex process than simply speaking of adoption suggests.

Discussion: Concepts for Understanding Adoption

Although this study sheds some light on adoption, clearly a more principled understanding is needed. In this section we discuss three concepts - critical mass, social affordances and interaction ecologies - that we have found useful in trying to understand adoption, and that we believe could benefit from further elaboration.

Critical Mass

Adoption is often thought to require a critical mass of users. While generally true, the adoption patterns we observed suggest a need to elaborate this relationship. For example, the Human Resources Babble seemed to us to be "fragile," for lack of a better term. Fragility can be framed as an issue of critical mass (Grudin, 1988; Markus, 1990), and indeed this Babble seemed barely to have a critical mass. With ten potential users, less than five posted in any given week (Figure 5). One or two people, by dropping out, might cause the deployment to fail. As we examined the nature of the messages in this Babble we noticed that with certain communicative practices, (e.g., information exchange), it was the composition of participants rather than the number of people involved in the practice, that reinforced use. For example:

> "Linda and I were very productively able to use Babble for some very specific things. So the critical mass there is two. For very hard and fast stuff. For a repository of information on something as important as BOV, it has to be the whole group." - Recruiter

Certain communicative practices, then, may require as few as two users to legitimate and sustain use of a groupware system. In much the same way that interface features (e.g., meeting reminders in shared calendars) provide individual benefit and may attract a critical mass of users (Grudin & Palen, 1995) productive communicative behaviors practiced among user dyads may reinforce adoption.

Furthermore, Babble's adoption patterns support the notion that use is an interactive phenomenon where payoffs to (a subset of) adopters may depend on the behavior of both adopters and non-adopters (Markus & Connolly, 1990). Consider a scenario where as the number of adopters increases, payoffs to non-adopters decrease, and payoffs to adopters increase. This situation exists when group members use Babble to keep each other 'in the loop.' If a preponderance of members are using Babble in this way, then non-adopters will be 'out of the loop' if they don't adopt. Since non-adopters are presumably motivated to stay 'in the loop,' it would seem that this practice would inevitably lead to adoption. But this is not necessarily the case. Assume that a group uses only one (electronic) medium at a time to 'keep in the loop', and that maintaining 'the loop' requires that all members have access to the same information. Then, from the perspective of any member, adoption payoffs exist under two conditions: when the entire group is using Babble to broadcast information, or when no one is using it (i.e., all use another medium). Critical mass in this scenario is an all-or-none phenomenon. Thus ironically, the communicative practice of 'staying in the loop' creates a condition where partial adoption is conceivably worse than no adoption! In effect, the critical mass for Babble depends on which communicative practices are enacted within a given group.

Social Affordances

We have discussed the ways in which the technical properties of the Babble interface enable certain types of communicative practices among its users. We believe there could be great value in framing this sort of relationship more precisely. One approach to this is to draw on the notion of affordances. Originally proposed by J.J. Gibson (1979), "affordance" refers to the relationship between an object's physical properties and characteristics of an agent that enable particular interactions between agent and object. Thus, a door knob affords grasping and turning to a normally-abled human, but not to one with severe arthritis or to one lacking an opposable thumb. The term affordance was appropriated by Norman (1988) and Gaver (1991) as a conceptual tool for discussing the design of interactive systems. It has proven useful for understanding how design and perception impact technology use, and it could be useful in explaining groupware adoption.

Our working definition of a social affordance [Note 1] is *the relationship between the properties of an object and the social characteristics of a group that enable particular kinds of interaction among members of that group*. For example, consider a door that opens out into a busy hallway. If a person opens the door quickly, it may strike someone entering from the other direction. One possible solution is to put a glass window in the door. The glass window addresses the problem at two levels. At the level of individual perception, the glass makes a person on the other side visible (i.e., the window affords seeing through it to a sighted person). At the social level, since people are socialized to not strike others with doors, they will refrain from doing so if given the chance. Furthermore, not only can the potential door opener see through the window, but the person on the other side can see as well, and thus there is shared knowledge of the situation (e.g., 'I know that you know that I know'). As a consequence, the door opener will be held accountable for her actions. This accountability, which arises from the optical properties of glass, human perceptual abilities, and the social rules of the culture, is an example of what we call a social affordance.

To apply this to the case of Babble, let's consider waylay through the lens of social affordances. Instances of the communicative practice of waylay - watching for a person to be active and then opening a communication channel with them - were observed in many groups. However, the existence of waylay does not mean that it is welcomed, necessarily results in helpful interactions, or is viable over the long term. For example, remote users in one group feared that their manager would use Babble to waylay them and delegate work to them, and hence avoided using Babble. Thus, although Babble supported the communicative practice of waylay, waylay was not socially afforded because of the social characteristics of the group. This stands in contrast to other groups in which waylaying was a regular and recognized feature of group interaction, and members might issues invitations to waylay, like 'catch me

waylaying was a regular and recognized feature of group interaction, and members might issues invitations to waylay, like 'catch me when I'm on Babble and we'll chat about it.' In these cases, waylay is socially afforded by the interaction of Babble'S functionality and group characteristics such as stronger social ties, generalized reciprocity, and perhaps by shared understandings of the limits of what may be asked in a waylay.

This discussion highlights one other useful feature of the concept of social affordances: the notion that an affordance is bound up with perception. In Gibson's original work, affordances were important because agents were seen to perceive objects (e.g. the doorknob) in terms of affordances, that is, in terms of potential interactions (grasping and turning). In our appropriation of the term, we suggest that as a group gains experience with a system, it comes to understand, collectively, how to appropriately apply the system to its own ends. That is, Babble's social affordances are the set of recurring communicative practices that a particular user group has come to recognize as appropriate and legitimate. While the concept requires more elaboration, particularly via application to a variety of cases, we believe that it may offer a principled way to discuss adoption.

An Ecology of Communicative Practices

Based on our observations of the different Babbles our intuition is that some deployments have, for lack of a better word, a sort of 'life' to them. That is, some instances of Babble seem alive and healthy; others seem alive but fragile; and others dead or dying. As we watched groups' reactions to Babble, our notion of 'aliveness' began to draw on an ecological metaphor. We began to think of the various Babbles as ecosystems populated by communicative practices requiring the input of energy from human participants to be sustained.

For example, compared to the Human Resources Babble, the Lab Babble is a very 'healthy' interaction ecology that is characterized by a great diversity of communicative practices (species). One common practice in the Lab Babble is the "Morning Greeting:" members of the lab often log in to Babble, say 'Good Morning' to one another and then engage in social chat about the weather, etc. Another practice is "Announcements:" lab members use Babble to announce impromptu meetings and issue reminders about events.

Looking closely at these practices, their relationships to one another and to the group engaging in them we notice three things. First, the way in which people engage in a communicative practice differs from practice to practice. Two people can indeed sustain the Good Morning activity, if they both 'speak.' However, even though a larger number of people is required to sustain Announcements, all that is necessary is that one person 'speak' and others 'notice' the announcements often enough to make use of them. Thus, different 'species' of communicative activities not only require a different number and different composition of participants, they require different types of participation. Second, it appears that some communicative practices are stronger attractors to participation than others. For example, it is our sense that although people make the effort to log on to Babble to read announcements (thereby 'staying in the loop'); they do not usually log on to Babble expressly to say 'good morning.' But, once connected for some practical reason, they are happy to participate in the 'good morning' practice. Third, we found that both dependencies and interdependencies of practices exist. Certain communicative practices, such as 'talking out loud' and peripheral monitoring, are interdependent: they reinforce one another. Others are dependent: although the 'announcements' practice reinforces the 'good morning' practice, the reverse is not true. If for some reason the announcements activity ceased, the 'good morning' activity might suffer, but the inverse is not likely.

While viewing CMC systems as ecologies of communicative practices is currently closer to metaphor than it is to a crisp conceptual framework, the work of Pirolli and his colleagues on information foraging (e.g., Pirolli & Card, 1995) demonstrates that ecology offers useful models that can be applied to other ends. It remains to be seen whether the adoption of CMC systems might be usefully modeled in this fashion.

Conclusions

We are not the first to report that local culture impacts groupware adoption. Orlikowski's (1992) and Bowers' (1994) studies are just two that draw similar conclusions. We have, however, approached adoption in a more systematic way by conducting a longitudinal study of several groups within the same organization.

Discussing controversy that arose surrounding the adoption of a CMC system, Bowers (1994) observed: "...several members felt that sending messages to a computer conference was a dangerous thing to do." Our study helps to answer questions such as: 'What *specific* features of a software interface (could) make it dangerous to use?' and 'Which social practices give rise to the perception of danger?' More generally, we have demonstrated how specific technical properties of a novel CMC system interact with the social characteristics of groups to enable a variety of communicative practices. These practices, depending on the ends to which they are put and their social context, in turn impact adoption.

We have also suggested a number of concepts that may be useful in future analyses of adoption. The concept of social affordances offers a way of getting a handle on the interplay between a communicative practice and the social characteristics of the users group, particularly with respect to practices that the group recognizes as legitimate. We believe that the notion of critical mass requires further elaboration, since our observations indicate that the number and roles of users necessary to sustain a communicative practice differ from one practice to another. One possible elaboration involves the concept of an ecology of communicative practices, a way of framing the complex dependencies and interdependencies among various communicative practices.

Our study's focus on chat is timely. Recent reports suggest that many US and European organizations are presently looking towards technologies based on chat to support distance collaboration in organizations (McGrath, 1999; Wojcik *et al.,* 1999). Yet most research in this area focuses on user modifiable virtual worlds (Doppke, 1998), virtual communities (e.g., McGrath, 1999; Wojcik *et al.,* 1999) or virtual, 3D environments (Greenhalgh & Benford, 1995). By closely examining the uses of chat in the workplace, this study may assist others in understanding the technical and social possibilities that chat offers for collaboration.

Acknowledgements

This research was conducted while the first author was a summer intern at the IBM T.J. Watson Research Center in 1998. This research has been partly supported by grants from the CISE/ISS/CSS Division of the U.S. National Science Foundation, and the NSF Industry/University Cooperative Research Center on Information Technology and Organizations (CRITO) of the University of California, Irvine. We thank Mark Ackerman, Amy Bruckman, Jason Ellis, Jonathan Grudin, Christine Halverson, and three anonymous reviewers for comments. Thanks to Teresa Wilkeson at UCI; and to the Social Computing and Niche Networking groups at Watson, particularly Cal Swart for critical work in the deployment of the Babbles, and David N. Smith, Mark Laff, and Amy Katriel, the Babble programmers.

Notes

1. First used but not defined in Ackerman & Palen, 1996.

References

Ackerman, M. and Palen, L. (1996). "The Zephyr help instance: Promoting ongoing activity in a CSCW system," *Proceedings of CHI '96*. ACM, New York, pp. 263- 275.

Adams, L., Toomey, L., and Churchill, E. (1999). "Distributed research teams: Meeting asynchronously in virtual space," *Proceedings of the 32nd Hawaii International Conference on System Sciences (HICSS'99)*, IEEE.

Bowers, J. (1994). "The work to make a network work: Studying CSCW in action," *Proceedings of CSCW'94, Conference on Computer Supported Cooperative Work*, pp. 287-297.

Churchill, E.F. and Bly, S. (1999). "Virtual environments at work: Ongoing use of MUDs in the workplace," *Proceedings of WACC'99*, February 22-25, 1999, San Francisco, CA.

Correll, S. (1995). "The ethnography of an electronic bar," *Journal of Contemporary Ethnography*, vol. 24, no. 3, pp. 270-298.

Doppke, J., Heimbinger, D., Wolf, A. (1998). "Software process modeling and execution within virtual environments," *ACM Transactions on Software Engineering and Methodology*, vol. 7, no. 1, January 1998, pp. 1-40.

Ehrlich, S.E. (1987). "Strategies for encouraging successful adoption of office communication systems," *ACM Transactions on Office Information Systems*, vol. 5, pp. 340-357.

Erickson, T., Smith, D.N., Kellogg, W.A., Laff, M.R., Richards, J.T. , and Bradner, E. (1999). "Socially translucent systems: Social proxies, persistent conversation, and the design of 'Babble'," *Proceedings of CHI '99* (Pittsburgh, PA May 15-20), ACM, New York.

Fish, R., Kraut, R. E., Root, R., Rice, R. (1993). "Video as a technology for informal communication," *Communications of the ACM*, vol. 36, no.1, January 1993, pp. 48-61.

Gaver, W. (1991). "Technology affordances," *Proceedings of CHI 1991* (New Orleans, Louisiana, April 28-May 2, 1991) ACM, New York, pp. 79-84.

Gibson, J. J. (1979). *The Ecological Approach to Visual Perception*. New York, NY: Houghton Mifflin.

Greenhalgh, C. and Benford S. (1995). "Massive: A virtual reality system for tele-conferencing," *ACM Transactions on Computer Human Interfaces*, vol. 2, no. 3, pp. 239-261.

Grudin, J. (1988). "Why CSCW applications fail: Problems in the design and evaluation of organizational interfaces," *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW '88)*, Portland OR, pp. 85-93.

Grudin, J. & Palen, L.: (1995) "Why Groupware Succeeds: Discretion or Mandate?," *Proceedings of ECSCW'95, European Conference on Computer Supported Cooperative Work (ECSCW '95)*, pp. 263-278.

Heath, C. and Luff, P. (1991). "Collaborative Activity and Technological Design: Task Coordination in London Underground Control Rooms," *Proceedings of ECSCW'92, European Conference on Computer Supported Cooperative Work*, pp. 65-80.

Kovalainen, M., Robinson, M., Auramaki, E. (1998). "Diaries at work," *Proceedings of CSCW '92* (Seattle, WA), November 14-18. ACM, New York, pp. 49-58.

Markus, M. L. (1990). "Toward a 'critical mass' theory of interactive media," In J. Fulk and C. W. Steinfield (eds.), *Perspectives on Organizations and New Information Technology*. Newbury Park, CA: Sage Publications.

Markus, M. L. and Connolly, T. (1990). "Why CSCW applications fail: Problems in the adoption of interdependent work tools," *Proceedings of ACM CSCW'90 Conference*, pp. 371-380.

McDonald, D., and Ackerman, M. (1998). "Just talk to me: A field study of expertise location," *Proceedings of CSCW '92* (Seattle, WA), November 14-18, ACM, New York, pp. 315-324.

McGrath, A. (1999). "The Forum: Helping people who cannot be together, work together," *SIGGROUP Bulletin*, vol. 19 no. 1, ACM, New York.

Norman, D. (1988). *The Psychology of Everyday Things*. New York: Basic Books.

O'Day, V.L., Bobrow, D.G., and Shirley, M. (1996). "The social-technical design circle. In ," M.S. Ackerman (ed.), *Proceedings of the ACM 1996 Conference on Computer Supported Cooperative Work*, pp. 160-169.

Orlikowski, W. (1992). "Learning from Notes: Organizational issues in groupware implementation," *Proceedings of CSCW'92, Conference on Computer Supported Cooperative Work*, pp. 362-369.

Pirolli, P. and Card, S.K. (1995). "Information foraging in information access environments," *Proceedings of the Conference on Human Factors in Computing, CHI '95*, (Denver, CO), ACM Press, pp. 51-58.

Sproull, L. and Kiesler, S. (1991). *Connections: New Ways of Working in the Networked Organization*. Cambridge, MA: MIT Press.

Turkle, S. (1995). *Life on the Screen: Identity in the Age of the Internet.* New York, NY: Simon & Schuster.

Watts, J., Woods, D., Corban, J., Patterson, E. Kerr, R., Hicks L.C.(1996). "Voice Loops as Cooperative Aids in Space Shuttle Mission Control," *Proceedings of CSCW'96, Conference on Computer Supported Cooperative Work*, pp. 48-56.

Wojcik, R., Fuchs, L., Poltrock S. (1999). "Business value of 3D collaborative virtual environments," *SIGGROUP Bulletin*, Vol. 19, No. 1, ACM, New York.

---

* To appear in *The Proceedings of the European Conference on Computer-Supported Collaborative Work (ECSCW)*, September, 1999.

[Tom's Home Page]
[Professional] [Life, Fun, &c] [Tell Me...]
