

What Counts as Success? Rhythmic Patterns of use in a Persistent Chat Environment

Christine A. Halverson, Thomas Erickson, Jeremy Sussman
IBM T.J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY USA
01 914 914 6659

{krys, snowfall, jsussman}@us.ibm.com

ABSTRACT

This paper presents a case study of a globally distributed work group's use of an online environment called "Loops." Loops is a web-based persistent chat system whose aim is to support collaboration amongst corporate work groups. We describe the ways in which the group turned the system's features to its own ends, and the unusual usage rhythm that corresponded with the team's varying needs for communication as it moved through its work cycle. We conclude with a discussion of design implications, and a suggestion that "community" may not always be the best way to think about groups' use of online systems.

Categories and Subject Descriptors

H.5.3 [Information Interfaces and Presentation]: Group and Organization Interfaces – Computer supported cooperative work, evaluation/methodology, synchronous interaction, asynchronous interaction, web based interaction

General Terms

Design, Experimentation, Human Factors

Keywords

Software development, virtual community, chat, instant messaging, persistent chat, CMC, CSCW

1. INTRODUCTION

For the last several years our research group has been engaged in designing online conversational environments aimed at supporting small- to medium-sized corporate work groups that are geographically distributed. Our aim is to design "socially translucent" systems—systems that provide a social context for interaction by providing cues about the presence and activity of participants. We have argued that such systems can, by taking advantage of the human ability to draw inferences from traces of activity, support the same social processes (e.g. imitation; peer pressure) that permit collocated work groups to function

effectively [6].

Our work has been embodied in two systems. The first-generation system, *Babble*, is a chat-like, conversation-centric system that differs from most other chat systems in two ways. First, it features a social proxy—a minimalist visualization that provides cues about the presence and activity of the participants. Babble also supports 'persistent' chat, text-based conversation that has the immediacy of chat or instant messaging, but in which the text persists across sessions, thus enabling either synchronous or asynchronous interaction. In a series of publications we've described the design of the system [5], and our experience with various deployments: [2, 7]. More recently we designed, implemented, and have begun deployment of a second generation successor to Babble called *Loops*. Loops is a web-based conversational environment that maintains Babble's socially translucent features, and adds new functionality that allows users to create and edit static text for the purpose creating group announcements, checklists, phone lists, etc.

Because of our experience with Babble (approximately 25 deployments over about four years) and Loops (approximately four deployments begun during the last year), we had come to view Babble and Loops as community environments, in much the same sense as Mynatt, et al.'s concept of network communities [15]. In particular, we came to view our deployments as *places* that were continuously inhabited, and we had developed a rule of thumb that the success or failure of a given deployment of a Babble or Loop could generally be predicted by whether activity continued or dropped to near zero after about the sixth week of the deployment.

It was against this background that, in the wake of a server crash, we were surprised to receive an urgent message from a Loops community that we assumed to have been dead for several months:

"Fargo Loops has been down since earlier this afternoon. We're having a build release on Thursday and so are using Loops heavily these last few days, so we'd really appreciate getting it back online. Can you help?"

While the Fargo community had gotten off to a strong start, with hundreds of messages per week over the first eight weeks, it had suffered a radical decline in activity such that eight weeks later it had less than one message per day. Given our cumulative experience, we had written Fargo off as a viable community. After restarting the server, we investigated this apparent return from the dead, and discovered an interesting pattern of usage.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Group '03, November 9-12, 2003, Sannibel Island, FL.

Copyright 2003 ACM 1-58113-693-5/03/0011...\$5.00.

Fargo's patterns of activity were 'punctuated:' long intervals of silence were interspersed with periods of escalating activity that came to abrupt ends. Intrigued by a usage pattern that was wildly different from our experience (and from what we had designed for), we resolved to investigate this group's usage of Loops more carefully, on the theory that exceptions can be sources of insight.

Thus, this paper is a case study of a distributed work group and its practices, and how both the characteristics of the group and the communicative demands of its task shape its use of our system. We begin by discussing relevant work. Next we describe the group, its practices, the tools it uses, and provide a brief description of the user interface and functionality of Loops. We then turn to the data, describing our methods, the descriptive data, and discuss the way in which Fargo turned the Loops system to its own ends. We conclude with some reflections on the implications of this case for systems design.

2. BACKGROUND

There is, of course, a large literature that is relevant to these issues. For purposes of this case study, we will restrict our discussion to studies of the long term use of computer mediated communication (CMC) technologies in the workplace; in particular we will focus on systems which support conversational interaction.

We will take it as a given—and we suspect that few members of this paper's audience will object—that the effective functioning of distributed groups, like their collocated counterparts, involves interactions that range from planned to opportunistic. Furthermore, the talk of work groups is focused not simply on the work itself, but includes discussion of how to coordinate work, when to schedule meetings, and the more social sort of talk that contributes to the formation and maintenance of trust, cohesion, and other social facets of the group. (For reviews in this area see [11]). In view of the nature of Fargo's work, we will add only that software development groups appear to be no different from other work groups. Thus, for example, in a one-year study of 13 software developers from four departments, Perry et. al. [18] found that over half the developers' time was spent in interactive activities other than coding, and that about 75 minutes per day were spent in unplanned interpersonal interactions such as requests for informal code reviews, questions about development tools, and general problem solving and debriefing sessions

Now we will turn to recent studies of the use of conversation-oriented technologies in the workplace. While few of these studies are of globally distributed groups, most involve communication amongst people who are distributed in some way. (And, as Bellotti and Bly point out [1], even collocated groups can have similarities to distributed groups when the members' tasks require them to be "locally mobile", and pull them out of earshot or 'eye glance' of their colleagues.)

Most notably, the increasing adoption of instant messaging (IM) technologies in the workplace has resulted in a number of recent studies of the way in which IM has been used to meet the needs of people immersed in large organizations. Nardi et al. [16] report on an ethnographic study of IM, noting that IM is used not just for information exchange, but is often used to check availability, and negotiate times, places and channels for work-

focused communication. Issacs, et al [12] logged thousands of workplace conversations among the users of their Hubbub system, and evaluated the nature and functions of the conversations. They found that the primary use of Hubbub was for work conversations, and that secondary uses were for simple, single purpose interactions (28%) and for scheduling or coordination (30%). Handel and Herbsleb [10], studied the use of a semi-persistent group chat system by six globally distributed work groups. They found that the system was primarily used for bursts of synchronous chat, and that the content was focused on work talk and negotiating availability. (The finding that the system was primarily used for synchronous chat seems a bit surprising, given that the groups were globally distributed and that in the system studied the groups chat transcripts persisted for about a day, thus affording the possibility of asynchronous conversation amongst the groups' members.)

There have also been studies of the use of other, non-IM conversation systems in the workplace. Churchill and Bly [3] report on the use of a MUD over the course of several years by personnel at Argonne National Laboratory. The Argonne MUD is used for both work and social talk, and supports opportunistic encounters, planned interactions, and coordination. They also report that many users left their MUD windows open all day, and/or used logging software to support asynchronous interactions such as message leaving. Reporting on a more persistent sort of system, Kovalainen, et al. [14] describe the use of an electronic diary in a Finish paper mill. Based upon an analysis of 3500 entries, they argue that the entries constituted dialogues within and between work shifts, and that these dialogues shared some characteristics with conversations used to coordinate work in face to face situations. Finally, in a six month field study of Babble's use by six work groups ([2]), we demonstrated a variety of usage patterns spanning the spectrum from focused work to coordination to social talk.

In summary, there is ample evidence that conversation-oriented systems are used, and are useful, in the workplace. They support planned work interactions, opportunistic interactions, scheduling and coordination work, and social talk. However, for all the utility of such systems in the workplace, we are aware of no studies that examine the *patterns* of system use over long periods of time. So far the focus has been on adoption and the rare case of longevity – witness the 6 year case reported in [3]—but not a repeating pattern of use.

3. THE WORLD OF FARGO

Fargo is a project at a large software company that is developing a tool to aid web application developers. As a group, it is a globally and organizationally distributed effort. Fargo began as a research project and is now the code name of the Research and Development team. The Fargo project also includes a number of associated Research collaborators, and the Product Team that will take on responsibility for the system, its support, and its further development as it becomes a product. As a consequence, Fargo's work 'site' consists of multiple physical and computer-mediated places. We first discuss the composition of the group¹,

¹ The names of the company, the group, and the individuals have been changed to protect their privacy.

the work practice they are engaged in, and the various tools they use to communicate and coordinate their activities. We then introduce Loops, and describe how it has come to be another place in which to interact.

3.1 The Group and Their Practice

When we first encountered Fargo in May 2002 the group consisted of 14 people spread among 4 sites (New York; North Carolina; Japan, and India). They had already begun to change from a research only effort to a development effort. Individuals' roles included programmers, managers, and testers. Their original core was co-located, with connections to other groups developed over time. As time passed and the project continued to develop, some people left while others joined the project, resulting in 28 people in 5 sites, including a set of collaborators in Switzerland.

As a project engaged in research and development, their work practices alternate among various patterns. Most members of the R&D team are concurrently working on both research and development tasks. The research tasks are longer term, higher risk, and may be terminated if they do not succeed. The development tasks are more closely tied to a specific development cycle—in this case that of the Fargo project.

While Fargo is not yet a product, it is approaching that point, with a general release in the offing. So far development phases have lasted between 3 to 6 months, with major code releases being in the 6 month range, and incremental build releases in the 1-3 month range. In addition, these participants are distributed across different areas of the company. As Fargo gets closer to product, other parts of Global Software are being called upon to ramp up product-related aspects of the project such as documentation, market research, and sales.

Not surprisingly, the group uses a number of different media for communication and coordination purposes. The primary means of synchronous contact is the telephone and instant messaging (IM). However both of these media have two problems. First, given the time zone incompatibilities, reaching someone "on the other side of the world" requires getting up very early or staying up very late. In addition, both media are ephemeral. (While the IM client can be used to save conversations, this was not sufficient for the group's needs, as it only produced a record for an individual.)

As a consequence, asynchronous communications channels (in which the message persists and, by default, is available to all who participated) become very important. Lotus Notes email is the primary means of communication. In addition, two Lotus Notes databases used to fulfill different functions: a document store, and an action item list. (Notes databases permit comments to be attached to items, and so can support annotation, status tracking, and limited threaded conversation.) The document database provides a persistent place to collect documents, as well as notes and extended textual entries that may develop into documents. A new idea may start with an email thread, but as it develops it will get moved into a document stored in the documents database. The other Notes database acts as a repository and a method for tracking action items such as bugs—however action items can also include short term deadlines for features, documentation and virtually anything that needs attention.

This is an overview of the group, their project, and the principal tools they use. The two factors that are the most salient for the subsequent analysis are the team's geographical distribution and their need for closely coupled coordination and communication. As much of this activity took place in Loops, it is time to take a closer look at the system in order to understand the features it provided.

3.2 A Quick Overview of Loops

A Loop consists of a set of user-definable places, each of which can contain a conversation, static text, and people, as well as interface elements for seeing who is present, viewing, navigating and modifying the environment.

The user experience is that people log on to a Loops server, and enter a particular Loop, appearing in "the Commons," the default place that appears in all Loops. If others are present, it is not unusual for greetings, banter and other sorts of social talk to occur. Participants may also move from place to place, reading conversations that have changed in their absence, contributing new comments, and 'bumping into' other users as they do so. Users often 'hang out' in the "Commons", backgrounding Loops while they do other things. Background sounds produced by various Loops activities, or the social proxy, can be used to monitor Loops activity.

Figure 1 shows a screen shot of the Fargo Loop. The callouts indicate the basic parts of the user interface:

1. The Social Proxy. The social proxy is an awareness component that depicts people as small colored dots. It provides a glimpse of how many are in the current place and how recently they were active (people move to the center when active, and drift to the edge over about 15 minutes of idleness).
2. The Persistent Chat pane. Each place has a chat pane in which those who are in the place can 'talk.' Since the chat text is time-stamped and persists across sessions, conversations may be synchronous or asynchronous. Time-stamps are local to the speaker, so variations in location are apparent.
3. Slide-out Tabs. Each place can contain up to three slide-out tabs that can hold publicly viewable and editable text and URLs. These are often used for check lists, agendas, URL lists, and so forth.
4. The Places List. The places list shows a Loop's places, and indicates when a place contains new information, and when people are in it. All Loops contain a "Commons" place; other places are created as needed.
5. The People List. The people list shows those currently logged into the Loop, and provides access to person-centered functionality (ranging from private chat to creating new user accounts).
6. The Bulletin Board. Each place has a public bulletin board that is viewable and editable by all those with access to the place. Bulletin boards are typically used for announcements.

As described elsewhere [2, 7], Loops and its predecessor Babble tend to be used continuously, over relatively long periods of time, as group gathering and discussion places.

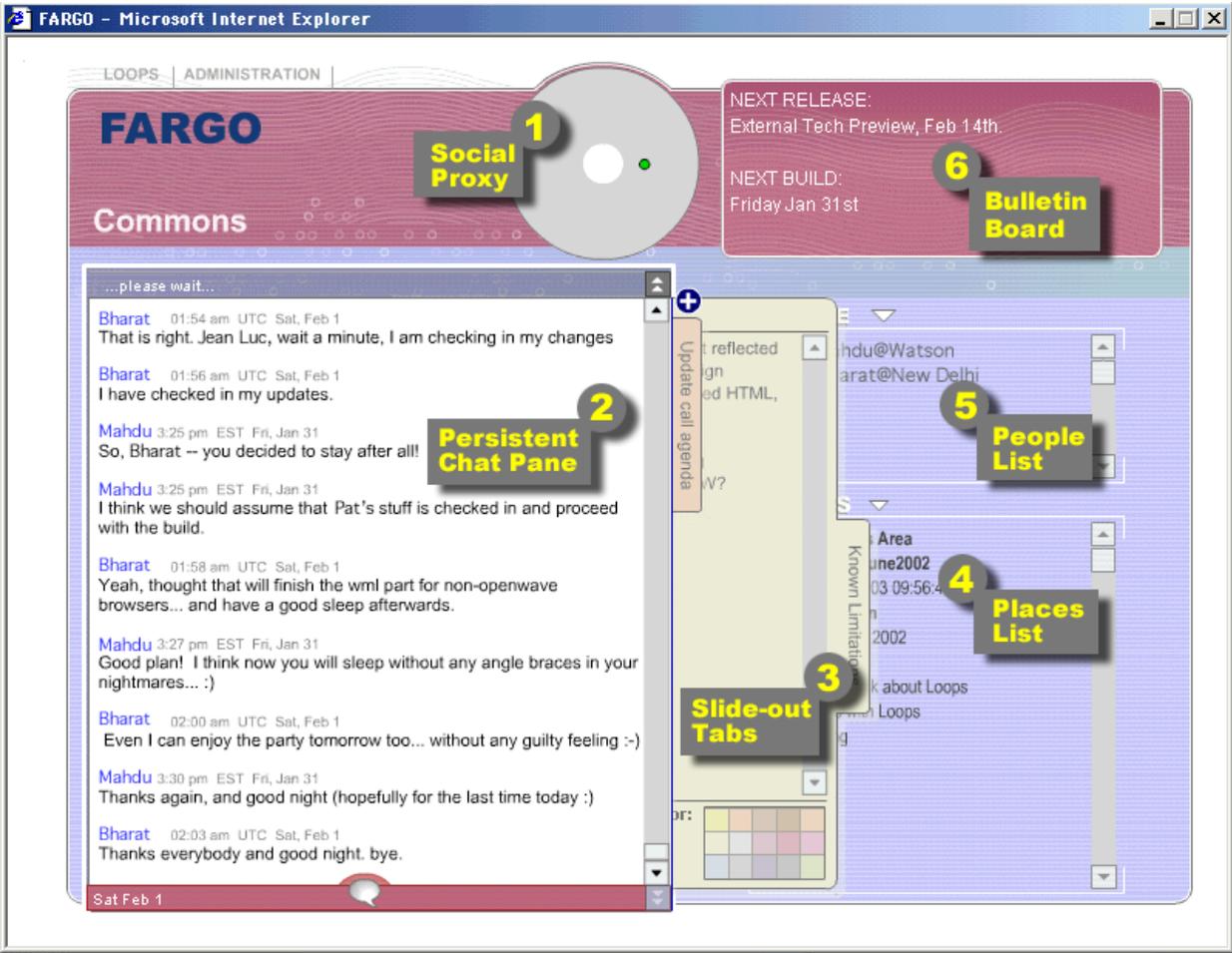


Figure 1. The Loops user interface.

4. DATA AND DISCUSSION

We deployed Loops to the Fargo team in May of 2002. Like most groups that we deploy to, early participation varied—in this case it was largely dependent on the presence of the manager. After the first 2 months Fargo continued to use Loops, but sporadically. After 4 months (Figure 2, bold line), when they dropped to less than three posts a day, we assumed that they were unlikely to survive as a viable deployment.

Thus we were surprised when we got an urgent message from the Fargo manager that the server had gone down and that they *really* needed Loops back up. We began to look more closely at Fargo’s use of their Loop and found a novel pattern of use. Rather than continuously inhabiting their Loop, they displayed a rhythm with increasingly intense use over a period of 2-5 weeks, followed by a drop off and an extended lull. In the following sections we describe the data we collected, its analysis, and what we found.

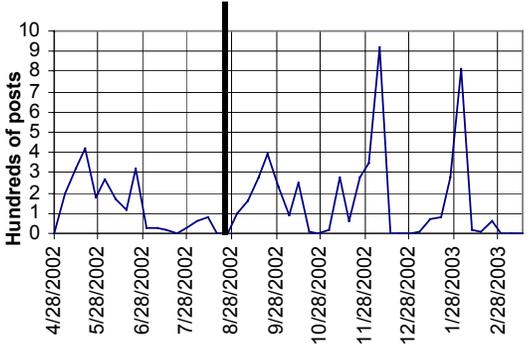


Figure 2. Posts per week. Bold line indicates point where we assumed deployment had “died”.

4.1 Study Methods

We used a combination of methods to study Fargo. First, we had surveyed the Fargo team before Loops was deployed. This was focused around their relationships with each other, and their knowledge of other team members. Second, Loops captures conversation, other authored content, and activity traces (the latter two via a logging mechanism). We analyzed the log files quantitatively tabulating records of logins, posts, idle times, and use of the tabs and bulletin board. We also analyzed the content of the postings in the chat pane and the tabs and bulletin board using two qualitative methods—coding categories as discussed below, as well as semantic understanding related to our knowledge of the field site. Third, we had access to contents of Fargo’s Document and Action Items databases. We skimmed their content and tabulated categories of entries for comparison

to the coded data from the conversation pane. Finally, we used semi-structured interviews check our analyses; we interviewed three Fargo team members who happened to be in our geographic area: the development manager, and two members of the original R&D team., all located in the Eastern U.S .

We focused our analysis on the content of two 2-week periods: one in December, 2002, and another in January-February, 2003. These were the most dramatic cases of the pattern of intense participation, and interviews indicated that they were important periods in the group’s development. (Less dramatic cases also occurred in August and October.)

As in previous analyses of Babble (and Loops), we were struck by the brief and informal character of the posts. The persistent conversation in Loops has the same casual feel and brevity that characterizes IM conversations. For this reason we took a closer look at recent analyses of IM and persistent chat [10, 12, 16].

We drew upon the work of Isaacs, et al [12] and Nardi, et al [16] in developing our coding scheme; however, there are several important differences. Nardi et al base their coding on self report of user categories. Isaacs et al coded IM ‘conversations,’ with a coding scheme derived from Nardi and augmented. We coded at the level of individual posts, because it is difficult to identify conversational boundaries in a persistent chat system. Each post could be coded as representing one or more categories; that is, a post might be coded as being an instance of both “work talk” and “social talk”, while another might only be “work talk”. When there were two clauses in a post that referred to work, the post was still only coded once for “work talk”.

The coding categories we developed are shown in Table 1. “Work talk,” of course, is focused on the group’s work; it includes, but is not limited to, quick question and answer pairs. “Work coordination” is coordination that is directly tied with the work practice (for example, the necessity of closely coupled interaction in debugging). “Scheduling and general coordination” cover more generic coordination tasks such as scheduling meetings, adjusting schedules, and so on. “Social talk” includes greetings, goodbyes, discussion of the weather, humor, and so forth. Perhaps because Fargo is a non-located work group, we do not see the more personal types of social talk reported elsewhere (e.g. couples arranging to get together for dinner [12]). Unlike the definition in Issacs et al, our definition of social talk follows Goffman’s notion of interaction rituals [8][9]. That is, the social glue of openings and closings that preserve face, mark replies, and so on; as well as addressing individuals by name, to put a more personal tone to the interaction. While we—and many of our readers—would argue that this *is* work, we coded them separately in order to see just how much time was spent on them. “Loops related” talk refers to discussion about the system; as we shall see, it was minimal.

Table 1. Content Coding categories

Work Talk (including Q & A)
Work Coordination (including debugging, etc.)
Scheduling and General Coordination
Social Talk (including greetings and good-byes)
Loops Related
Media Switches (request to change between media)

Finally, the “Media switches” category covers talk about whether or when to switch to another medium (such as phone), something observed in some studies of IM, notably [16].

All Loops content for these two 2-week periods were coded using this coding scheme. Both databases were examined for the same periods and categorized based on the group’s self defined categories: Discussion, Meeting Minutes, Reference, and Builds.

Having described our methods, we now move into a discussion of our findings. We will structure our discussion in terms of two general questions: How did the members of Fargo turn Loops to their own ends? And what explains the team’s punctuated usage pattern—what were they doing during the lulls?

4.2 Fargo’s Use of Loops - Structure

What were the members of Fargo doing when they used Loops, and how did they turn it to their ends? We will begin by looking at the structure and content of the conversation in Fargo Loops, and then examine the ways in which they used the static text areas in tabs and bulletin boards. As we present the descriptive data, we will occasionally pause to discuss the more general issues it raises.

4.2.1 Turn Size

We begin with a look at turn size, identifying conversational turns within posts in keeping with the more in depth analyses of IM [10,12]. In Fargo Loops, each turn averaged 10 words (ranging from 1 to 29), comparable to the 13.5 word average reported in Isaacs, et. al. [12]. As they point out, the brevity of each turn is not incompatible with other estimates of workplace interactions, whether F2F or mediated through a telephone or other means.

4.2.2 The Social Structure of the Chat

In terms of the social structure of the talk, we are limited in what we can infer about the fine structures of the interaction. Loops display of conversation is broadcast, so we can only identify conversational partners with certainty when a post is addressed to someone (e.g. “Bob--have you checked in your code yet?”), something that does not happen regularly.

However, we can examine the relative dominance of speakers. We find that, in Fargo, the development manager, Madhu, plays a key role. A reading of the posts for these periods showed us

Table 2 Breakdown by Person comparing CMC use

	December Posts			January-February Posts	
	Loops	TeamRoom		Loops	Team Room
Madhu	551	8	Madhu	382	4
Bharat	116		Billy	143	0
Ichiro	98	3	Pat	134	4
Jean Luc	98	1	Jean Luc	109	3
Junko	86	2	Bharat	105	1
Catherine	85		John	75	6
Junji	67		Catherine	60	1
Billy	57		Elizabeth	14	0
OTHER	102	5	OTHER	65	9
TOTAL	1262	20	TOTAL	1087	32

that everything seemed to revolve around him. A closer look confirmed this (see Table 2).

Madhu had the most postings in both periods, as well as the most uses of tabs and bulletin boards. In our interview with him, Madhu attributed his degree of participation to his need to track everything, making sure problems were fixed in a timely manner, particularly in the week leading up to the first external release of the beta-version of Fargo. Madhu emphasized that he often felt that he was really pushing to get people onto Loops, but that it made coordination so much easier for him, it was worth it.

Our interview with Madhu led us to expect that other Fargo team members had a certain amount of resistance to the use of Loops. For example, another long term group member that we interviewed, Billy expressed a number of reservations about using Loops. He cited memory drain on his development environment, the need to remain focused, and the difficulty of monitoring without getting side tracked, leading us to the expectation that he was a minor participant. In contrast, we found that Billy was one of the top 8 posters in December and the second highest poster leading up to the February release. (See table 2.)

4.2.3 Lurking and Listening

What we have not discussed is how many people logged in to the Fargo Loops during this period, not to post, but to read. Such persons are often referred to as “lurkers,” a term that we have some problems with.

Much of the CMC literature focuses on posting as a means for measuring the success of a particular technology, or if a MUD, MOO, or such, of a community. The term “lurker” has come to have negative connotations. Kollock and Smith [13] have included lurkers among free-riders—that is, members who use resources but do not contribute. From this point of view lurking needs to be minimized in order to ensure a successfully working community. On the other hand, Nonnecke and Preece [17] argue that lurking is normal, and should be seen as a valid form of participation that can, in some cases, add value to the group.

In the case of Fargo Loops, it seems evident that lurking is a valid form of participation. There are two reasons for this. First, the Loops social proxy, by representing anyone present as a name and a dot, turns lurkers into a visible audience—something which anyone who has ever given a talk to a nearly empty room can appreciate. Second, as a communication nexus for an active work group, the fact that people are following the discussion—even if they don’t contribute—adds value by supporting shared awareness of the project’s state. In fact, lurking—or listening—is quite common in Fargo Loops. Of the total number available to participate, two thirds are listeners. In comparison with Discussion Lists this is a high rate of participation. However, we might still ask why it is not higher.

One reason has to do with cultural and associated pragmatic issues. The Japanese team members don’t use Loops to talk with each other: they are using machines with keyboards that make typing among themselves in English cumbersome.² Yet several of them log on and maintain a presence on the Loop, reading about what has happened while they were asleep. Some of these

² This was reported to us second hand in two interviews.

participants never post, while others occasionally post in the times they overlap with those in the United States, 12 hours away. In contrast, Bharat may have a more prominent presence because, at the time, he was the only team member in India. For him Loops is a vital means of staying in touch with the rest of the team.

Another source of lurkers are the new people who joined Fargo to support the productization effort. Their posting rate was low (some never posted), perhaps because Fargo’s primary use is to coordinate development, but they began to hang out in Loops to keep abreast of the development process. But, as they prepared for the internal release in February, the product team’s posting picked up, as they coordinated the documentation process. In each case, the persistence of the talk provided the means to share decisions and status with partners on the other side of the world without having to write additional email.

4.3 The Content of the Persistent Chat

Table 3 shows the frequencies of the primary coding categories (note that the two coordination coding categories are combined, and media switches will be treated later).

Table 3. Incidence of categories (% of posts). Posts may be multiply categorized so rows do not sum to 100

	Work Talk	Coordination Talk	Social Talk	Loops related
Dec	84.3%	19.1%	19.7%	1%
Jan/Feb	81%	13.6%	21%	5%

Not surprisingly the majority of the talk (above 80%) was work related. As noted in other studies of CMC in the work place, a significant portion of time is spent in both social talk and coordination. Coordination talk was perhaps more extensive than reported in Isaacs et al. [12], although since our percentages are coded against posts, while theirs are by conversation, the comparison isn’t exact. Social talk hovered at around 20%, and talk about Loops was quite rare.

4.3.1 A Closer Look at Coordination Talk

Coordination talk in this case seemed to be more than setting up meetings. As Table 4 shows, we broke out coordination into the

Table 4 Breakdown of Coordination Talk and Media Switches (% of posts)

	Work Coord	Sched. & Coord	Media Switches	Negative Switches
Dec	8.6%	10.4%	0.6%	0.2%
Jan/Feb	7.2%	6.4%	1%	0.6%

tightly coupled work interactions required for debugging and negotiating code check-ins and builds, as well as the more expected scheduling and coordination tasks. We found the relative frequencies fairly evenly divided.

The excerpt below shows the kind of closely coupled coordination we observed and coded as work coordination.

Madhu 11:56 am Folks -- it's time to checkin your code so we can start the build.

Madhu 11:56 am If you're making documentation changes, you can continue that, but we should start testing the code build.

Madhu 11:58 am Please everyone acknowledge that you've checked in your code changes.

John 11:59 am ack

Billy 12:00 pm done... for now

Madhu 12:01 pm (Billy, we can incorporate your bug fix later...)

Madhu 12:06 pm Catherine ack'ed to me privately.

Pat 12:06 pm I'd like about three more minutes to finish adding some trace points

This coordination is a strange cousin to the local mobility in Bellotti and Bly [1]. In this case, even though many of the individuals are on the same floor, people are out of touch because of local *immobility*: they have to be at their workstations to check in their code. While they could acknowledge by phone, having the persistent record acts as a kind of *party line*, informing everyone of the status. The frequency of this kind of coordination was relatively equal to the more expected scheduling reported elsewhere.

4.3.2 Media Switches

Table 4 also shows that we didn't find very many Media switches. (Note that the percentages here are of incidents per total number of posts, so it is not directly comparable to Isaacs et al.'s [12] per conversation measures.) Media switches are defined as a request to change to another media, often phone. Nardi [16] suggests that as things get too complicated for IM a media switch happens. Isaacs et al. [12] in contrast found fewer media switches than indicated by self report. Following Isaacs, we also find the incidence of media switches to be low. The conversations were rapid fire and short, although of some technical depth (particularly for debugging). Thus it was difficult to evaluate evidence of switching because of increasing complexity.

What we also found is what we term *negative* media switches. These were cases where people were enjoined to stay *on* the Loop rather than switch to another media. We saw three rationales for these switches.

1. Failure of another medium, such as slow email.
2. The persistence of the text provided a common location for others to catch up.
3. The efficiency of synchrony was necessary because of time constraints of the impending release.

For example, as they are finalizing documentation they are waiting for a piece of information from John. However, the team has not yet received it. In the conversation below we see they turn to pasting it into the Loop as an alternative to the failed email.

Billy 3:30 pm John, did you replicate Notes?

John 3:29³ pm Only 3 times since then....

John 3:30 pm make that 4 times

Billy 3:31 pm not enough :) -- can you re-send it?

John 3:30 pm

John 3:32 pm resent and re-replicated..

³ Time stamps are based on the client's clock, so apparently out of sequence items reflect variations in user's clocks.

Billy 3:33 pm I'm looking at my Notes and see that I'm getting some mailed delayed by 30 minutes or more. Here we go again!...

Madhu 3:33 pm John -- one thing I need is to do a "smoothing pass" on the Samples, Tutorials, and References sections.

Billy 3:34 pm John, is it a short list you can copy-paste here?

In the next example, under deadline pressure, we see that because Jean Luc and Billy are co-located in New York, and Bharat is in India, it is faster and cheaper for Jean Luc to contact Billy for Bharat. It also is more efficient for the project, rather than relying on email.

Bharat 11:56⁴ pm anyway, will send a mail to Billy and wait for his help

Jean Luc 1:27 pm I can ask Billy to connect on loops if you want

Bharat 11:58 pm That would be great

Jean Luc 1:30 pm I've just gave him a call

Bharat 00:00 am Thanks Jean Luc

Bharat 00:00 am Hi Billy

4.4 Other Content – Tabs and Board

One of the most interesting changes from Babble is that Loops provides group text areas in the bulletin board and tabs. This text persists, but its persistence is different from that of the Chat pane. The chat appears to be dynamic due to the interplay of posts; however, once posted, individual posts don't change. In contrast the bulletin board and tabs *appear* more static (and in practice their content often remains static for long periods), but anyone can change their contents. Another aspect of the bulletin board and tabs is that, drawing upon activity logs, it is possible to make inferences how frequently each is read. With the chat pane we can only assume that anyone in the same room has 'heard' (i.e. read) the conversation. However, the bulletin board is larger than it appears in Figure 1, and for any substantial post the user must click on the board to expand it and read the whole text. Similarly, tabs are completely hidden, again requiring someone to open them, if they want to read the content. Such 'reading' behavior can be seen in Loop's logs.

In the first build cycle we saw extensive use of the bulletin board. Over the one week period that we looked at closely we see the bulletin board being used for announcements, with regular updates 2-3 times per day. All but two of these updates were created by Madhu. (In contrast the only use of the tabs during this period was by Dick⁵ who created two tabs, but did not change the default content of the tabs.)

The period starts out with the announcement of the final dates for the build and the internal preview release. (For example, see the bulletin board in Figure 1). Madhu opens and closes the bulletin board repeatedly, only making one change. That night he adds *Burning Issues*— issues that need to be fixed before the preview can go out. Each day he adds and subtracts from the burning issues, and advances the build date, day by day. The day before the final build he begins leaving the burning issues up

⁴ Some time zones in India are offset by half an hour from European and North American zones.

⁵ Dick is part of the product team.

there and noting in parentheses whether they are complete or not. Late on the 12th (NY time), Junji in Japan (mid day) marks one burning issue complete.

Early the following morning, Madhu announces that the internal preview is ready to go and the final build is on the database. He posts in the text that the Japanese testing team has signed off on the build. After another 20 minutes, where Madhu has opened and closed the BB many times without change, he adds “Congratulations” and thanks the team. An hour later (8:20am , 10:20pm Japan) Junji posts that the build is now on the Japanese test team’s web site, and the traffic winds down.

The next build cycle shows a more sophisticated use pattern, alternating tab and bulletin board use. In addition, there are a number of differences in terms of what is being posted, who is posting, and who is looking.

As before Madhu announces the deadline for the next release and the next build in the bulletin board. He also opens one of the tabs that is already created and changes it to list 5 key issues for this build. As before, build dates are updated on the bulletin board. However, the other information has moved to the tabs. Two hours after Madhu announces the build and release dates, he changes the date of the external preview release. Later that day, two other members of the team open and close the bulletin board without making any changes. The next day, Billy changes the release date by a week. Three days later Madhu changes the build date.

In contrast with the previous build-release cycle, almost all of the more detailed information is being tracked in tabs. These details include key issues for the build, known problems, and information for documentation. About midway through this cycle the key issues became “action items” which were then also transported to (and from) the Action Items database, often as HOT⁶ items.

One of the most interesting patterns is who reads the tab entries. While we see the development team writing to the bulletin board and the tabs, we primarily see members of the product management team reading them.

4.5 Archiving

Although Loops does not provide any facility for archiving material, the users of Fargo Loop appropriated a feature of Loops to serve this end.

Loops makes it easy for users to start new places. The rationale is that different places will be used for conversations focused on particular topics (and we have observed such behavior in other Loop and Babble deployments [2, 5, 7]. However, the Fargo team didn’t create places for special types of conversation; instead they congregated in the Commons and had almost all their conversations there. This unexpected use of Loops produced an unanticipated side effect.

The first several versions of Loops used by Fargo (Fargo was an early adopter of Loops) had a performance problem connected with conversation size. As a conversation’s size increased (as it did with Fargo’s exclusive use of the Commons), load time increased dramatically.

As a workaround, after the Commons conversation grew to a point where performance suffered, Madhu would create a new room and would transfer the content of the commons into it. His aim was to create a conversation archive that would be there when he or anyone else wanted it (including for ISO reporting requirements). Since, on being launched, Loops only loaded the content in the (newly emptied) Commons, this greatly increased performance.

Although this archiving practice was in response to a bug, it underscores the value that Madhu placed on persistence. Other incidents also reinforced the value of persistence. In one case, as a side effect of a server change, it appeared (erroneously) that a number of the Fargo archives had been lost—this elicited a very concerned reaction from Madhu. Madhu’s emphasis on the value of the persistence of archives was echoed, to a lesser degree, in the other interviews. In yet another indication of the importance of persistence, Madhu noted the need for an index and for a search mechanism, as did another member of Fargo, albeit for different reasons.

4.6 When Fargo isn’t Using Loops

As we’ve noted, Fargo’s use of Loops was punctuated: periods of activity alternated with lulls during which almost nothing happened. In this section we turn our attention to the lulls, and ask what was happening, and why.

4.6.1 Activity During the Lulls

In interviews, individuals reported that these lulls in Loops activity coincided with a greater focus on development coding. (These periods of developmental coding preceded the more intense and iterative pre-release coding that occurred right before a “code drop”.) During developmental work, Fargo team members reported wanting to focus on their own coding, and decreasing their use of synchronous communications media (e.g. phone and IM) because they found them to be distracting.

We can see data consonant with these reports in Table 5. It shows how usage of the three communications media to which we had access varied over time (rows with gray backgrounds indicate lulls). Team members reported a preference for email during the developmental coding periods, and we can see that the use of the Team Room database (for posting Build Images, among other things) stayed pretty steady across the lull periods (excepting Christmas break). In contrast, when activity shifted into the intense pre-release phase of coding, the increased

Table 5 Breakdown Comparing CMC use
(gray shading indicates lull periods in Fargo Loop)

start	end	# wks	POST per CMC		
			Loop	Team Room	Action Item completed
8/18	8/31	2	4	43	7
10/20	11/9	3	30	55	10
12/1	12/14	2	1262	20	34
12/15	1/11	4	10	8	9
1/26	2/8	2	1087	32	22
2/9	2/22	2	28	31	6

⁶ In the Action Item database HOT means highest priority.

activity in Loops was accompanied by increased use of the Action Items database, presumably representing the increased need to wrap things up in the intense pre-release phase of coding.

Thus, the usage patterns of the various media reflect the pressures of the development cycle in its different stages. Immediately prior to a release, the demands of code integration coupled with increased time pressure result in a greater need for closely coupled interactions needed to finalize the code. In the lull periods, team members eschew distraction in the form of Loops, IM and telephone calls, and the postings of design documents and more extensive discussions carried out in asynchronous media increase. Global coordination among the team continues, but switches to the venue that best fits the team's need.

4.6.2 Reading During the Lulls

Despite our contention that lurkers are participants too, our analysis is predominantly focused on the peak periods, equating participation with posting. However, logins, as well as our ability to track the opening of tabs and bulletin boards, at least allows us to draw some inferences about reading behavior during the lulls.

Let's walk through the four lull periods identified in Table 5. The one in August is when we thought the deployment had died. During this time period we only see 4 posts, all by the same person, and only one other person logging in twice. The posts are by Madhu, the lurker is from the product team. The lull in October/November is more complex. The participation consists of "Hi – how are you", while numerous people from remote locations login and hang out without ever saying anything. Bharat consistently logs in, even when he does not post. The Christmas holidays have little activity. No posts for 2 weeks and then a very few. However, even in the period where there are no posts at all, there are still 32 logins with an average idle time of one and a half hours. In this case it seems as though people are hanging out in case anyone else pops in, but not until after the new year does that happen. At that point, though there are only 10 posts in the chat pane, we begin to see activity in the tabs and on the bulletin board, and an increasing presence reading them—spread across developers and product team members. Finally, immediately following the peak reported in February, we again see this pattern of low posts in the chat pane, but some continuing activity in the bulletin board and tabs. Despite appearances of inactivity, people (many of them members of the product management team) are still connecting and apparently accessing information.

5. CLOSING REMARKS

In this paper we have examined an unusual long term pattern of Loops usage, that contrasted sharply with the steady-state 'inhabited' pattern that characterizes other deployments of Loops and Babble. The punctuated pattern of Fargo Loops initially led us to believe that the deployment had failed; subsequent experience disproved this, and our analysis of the Fargo's usage demonstrates a number of ways in which they used Loop's features to achieve their ends (sometimes in accord with our expectations, and sometimes not).

The Fargo team continues to find their Loop to be an important part of their repertoire of communication tools. The mostly

synchronous character of the conversation is deceptive. Its synchrony has more to do with the closely coupled nature of the work and coordination necessary for that work than any indication that it is not supporting asynchronous use. In fact, logon patterns during quiet periods (both long duration lulls and the shorter night-time-in-the-US periods) indicates that other team members are using Loop's persistence to "stay in the loop" with the development process.

Success, then, must be measured, not in terms of the quantity of posts or logins, but with regard to how well the system supported the team's work. Loops supports some work well along a number of time scales. We see this in the coordination work in the posts, the information posted in the bulletin boards, and the 'to do'-like nature of the tabs.

In the lulls in Loops traffic we see evidence that Fargo's work practice is shifting as well. They have face to face meetings, conferences, and they spend more time working through design issues – recording both the discussion and the results in their TeamRoom. Despite the apparent shift away from the Loop, team members are still logging in and hanging out, and updating the more static seeming elements like the bulletin board.

All of these patterns suggest some implications for designing conversation-oriented systems for globally distributed work groups.

First, we believe the sort of activity described here makes a compelling case for the importance of persistent chat. Although the members of Fargo had access to instant messaging, the efforts the group went through to move into Loops for the final weeks before release indicate that synchronous chat was insufficient. As we note in the discussion, the persistence of chat played a number of important and useful roles. Further, the persistence of the bulletin board and tabs provided a communication role for those less actively engaged in development.

Second, the punctuated use of Loops suggests a number of ways in which Loops, or similar environments, might be redesigned. First, it seems clear that if Loops is not a continually inhabited place where community members habitually convene on a daily basis, there need to be ways of summoning a group to re-inhabit Loops. Such a mechanism might be automatic (e.g. notifications of when the amount of new material in a Loop passes a preset threshold), or, more likely, it might be socially mediated, i.e. a means by which someone like Madhu might invoke to reconvene the group.

A second implication of punctuated use is that we need to think more carefully about how content is transferred into and out of the system. In the case of Fargo, members did indeed transfer information into and out of Loops (e.g. between the Notes Action Database and the Loops tabs). However, Loops, as a system, provided no means to facilitate that—users had to simply rely on individual cut and paste of text. Much more could be done here.

A third implication of punctuated use has to do with the issue of archiving and history. That is, in a Loop that is continuously inhabited by a community, history is a continuous thing—an unbroken stretch of conversation gradually turns into history. With punctuated usage in contrast, interactions form episodic chunks, that are largely—though not always—irrelevant to the

present episode. In addition, the bulletin boards and tabs often support the chat. To us this suggests a need for a simpler archiving mechanism that permits past episodes to be packaged, indexed (so they may be searched if needed, as requested by Madhu the manager in an interview), and archived. This would make them available and useful, but ‘out of the way’.

The title of this paper asks “What counts as success?” We hope that we’ve established that the Fargo Loop is a success, in spite of the fact that it ‘failed’ our rule of thumb for identifying viable deployments (continuous activity after the six week mark), and in spite of its ongoing deviations from our expectations that a successful Loop is a ‘place’ that is continuously inhabited by an active community.

Perhaps the bottom line for designers of online systems is that we should be cautious about thinking in terms of community. While in the physical world virtually all communities are bound to places or locales, it is worth remembering that the converse is not true. As designers, we may be able to create the equivalent of online places, but as this case study demonstrates, how they are used — whether they are inhabited and “settled,” simply used as convenient ad hoc resources, or somewhere in between — depends on the group and their practices. We should not be too quick to apply success criteria gleaned from what, after all, is simply a compelling metaphor. Elsewhere we have argued [4] that “community” may not be the best, and certainly is not the only, framework for thinking about online systems for groups; this study proves to be a compelling reminder of that.

ACKNOWLEDGMENTS

Thanks to colleagues in the Social Computing Group — Wendy Kellogg, Catalina Danis, Denise Edwards, Jason Ellis, and Tracee Wolf. Thanks also to Mark Laff, a former member who keeps on programming on Loops; to Cal Swart, who keeps our servers running; to Robert Bowdidge who performs last minute data crunching; and to Jim Herbsleb who provided some helpful pointers into the literature. And thanks to the Fargo team.

REFERENCES

- [1] Bellotti, V. and Bly, S. (1996). Walking Away from the Desktop Computer: Distributed Collaboration in a Product Design Team. *Proceedings of CSCW '96*. ACM Press, pp. 209-218.
- [2] Bradner, E., Kellogg, W., and Erickson, T. Bradner, E., Kellogg, W., & Erickson, T. (1999) The Adoption and Use of Babble: A Field Study of Chat in the Workplace. *Proceedings of ECSCW '99*. Kluwer, pp. 139-158.
- [3] Churchill, E. F. and Bly, S. (1999) Virtual environments at work: ongoing use of MUDs in the workplace. *Proceedings of the International Joint Conference on Work Activities Coordination and Collaboration*. pp. 99-108.
- [4] Erickson, T. (1997) Social Interaction on the Net: Virtual Community as Participatory Genre. *Proceedings of the Thirtieth Hawaii International Conference on Systems Science*. (eds. J. F. Nunamaker, R. H. Sprague) Vol 6. IEEE Computer Society Press, pp. 23-30.
- [5] Erickson, T. Smith, D. N., Kellogg, W. A., Laff, M. R., Richards, J. T., and Bradner, E.. 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*. ACM Press.
- [6] Erickson, T. and Kellogg, W. Erickson, T. and Kellogg, W. (2000) Social Translucence: An Approach to Designing Systems that Mesh with Social Processes. *Transactions on Computer-Human Interaction*. Vol. 7, No. 1. ACM Press, pp. 59-83.
- [7] Erickson, T. and Kellogg, W. Knowledge Communities: Online Environments for Supporting Knowledge Management and Its Social Context. (2003) *Sharing Expertise: Beyond Knowledge Management*. (eds. M. Ackerman, V. Pipek, V. Wulf). MIT Press, pp. 299-325.
- [8] Goffman, E. (1967, 1982) *Interaction Ritual*. New York: Pantheon Books.
- [9] Goffman, E. (1981/1995) *Forms of Talk*. Philadelphia: University of Pennsylvania Press.
- [10] Handel, M. and Herbsleb, J. D. (2002) What is chat doing in the workplace? *Proceedings of CSCW 2002*. ACM Press, pp. 1-10.
- [11] Hinds, P. and S. Kiesler, S. (eds). *Distributed work*. MIT Press, 2002.
- [12] Isaacs, E., Walendowski, A., Whittaker, S., Schiano, D. and Kamm, C. (2002) The Character, Functions, and Styles of Instant Messaging in the Workplace. *The Proceedings of CSCW 2002*. ACM Press, pp 11-20.
- [13] Kollock, P. and Smith, M. A. (1996) Managing the Virtual Commons: Cooperation and conflict in computer communities. *Computer Mediated Communication: Linguistic, Social, and Cross-Cultural Perspectives* (ed. S. Herring). Philadelphia, John Benjamins Publishing Co., pp. 109-128
- [14] Kovalainen, M., Robinson, M., Auramäki, E. (1998). “Diaries at work,” *Proceedings of CSCW '92* (Seattle, WA), November 14-18. ACM Press, pp. 49-58.
- [15] Mynatt, E. D., Adler, A., Ito, M., and O’Day, V.L. Design for Network Communities. (1997) *Human Factors in Computing Systems: The Proceedings of CHI '97*. ACM Press, pp. 210-217.
- [16] Nardi, B., Whittaker, S., Bradner, E. (2000). Interaction and Outeraction: Instant Messaging in Action. *Proceedings of CSCW 2000*, ACM Press, pp. 79-88.
- [17] Nonnecke, B. and Preece, J. (2000) Lurker Demographics: Counting the silent. *Proceedings of CHI 2000*. ACM Press, pp. 73-80.
- [18] Perry, D. E., Staudenmayer, N. A., and Votta, L. G. (1994) People, Organizations, and Process Improvement. *IEEE Software*, July 1994, pp. 36-45.