TeleNotes: Managing Lightweight Interactions in the Desktop

STEVE WHITTAKER, JERRY SWANSON, JAKOV KUCAN, and CANDY SIDNER
Lotus Development Corporation

Communication theories and technology have tended to focus on extended, formal meetings and have neglected a prevalent and vital form of workplace communication—namely, lightweight communication. Unlike formal, extended meetings, lightweight interaction is brief, informal, unplanned, and intermittent. We analyze naturalistic data from a study of workplace communication and derive five design criteria for lightweight interaction systems. These criteria require that systems for lightweight interaction support *conversational tracking, rapid connection,* the ability to *leave a message, context management,* and *shared real-time objects.* Using these criteria, we evaluate existing interpersonal communications technologies. We then describe an implementation of a system (TeleNotes) that is designed to support lightweight interaction by meeting these criteria. The interface metaphor allows communications to be based around desktop objects, resembling "sticky notes." These objects are also organized into "desktop piles" to support conversational threads and provide mechanisms for initiating real-time audio, video, and application sharing. We conducted informal user testing of several system prototypes. Based on our findings, outstanding issues concerning theory and systems design for communication systems are outlined—in particular, with regard to the issue of managing conversations over time.

Categories and Subject Descriptors: H.1.2 [**Model and Principles**]: User/Machine Systems—*human factors;* H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems—*evaluation/methodology;* H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*evaluation/methodology;* H.5.3 [**Information Interfaces and Presentation**]: Group and Organizational Interfaces—*asynchronous interactions; evaluation methodology; synchronous interactions; theory and models;* I.3.6 [**Computer Graphics**]: Methodology and Techniques—*interaction techniques*

General Terms: Human Factors

Additional Key Words and Phrases: Audio, awareness, computer-media spaces, conversation management, impromptu communication, informal communication, interpersonal communications, lightweight interaction, mediated communication, remote collaboration, task management, video

Authors' addresses: S. Whittaker, ATT Labs—Research, 180 Park Avenue, P. O. Box 971, Florham Park, NJ 09732; email: whittaker@research.att.com; J. Swanson, J. Kucan, and C. Sidner, Lotus Development Corporation, One Rogers Street, Cambridge, MA 02142.

1. LIGHTWEIGHT COMMUNICATIONS

Studies of workplace activity show the *pervasiveness* of interpersonal communications. Office workers spend between 25% and 70% of their time in face-to-face conversations with others, depending on job specification [Kraut et al. 1993; Panko 1992; Whittaker et al. 1994]. In addition to being frequent, interpersonal communication is also *vital* for achieving certain types of work-related tasks. Frequent, opportunistic face-to-face communications are crucial for rapidly resolving ambiguity during the planning and negotiation phases of projects, and they support organizational learning [Finholt et al. 1990; Kraut and Streeter 1996; Kraut et al. 1990; 1993; Suchman and Wynn 1984]. However, apart from a few recent exceptions [Bly et al. 1993; Fish et al. 1992; Gaver et al. 1992; Isaacs et al. 1996; Tang et al. 1994], past systems and theoretical work have tended to focus on only one class of interpersonal communication. This is the class of interactions that are *extended, multiparty, formal,* and *one-shot* [Egido 1988; 1990; Johansen 1984; Nunamaker et al. 1993; Olson et al. 1992; Stefik et al. 1987; Tang 1991]. Recent research shows, however, that such extended, multiparty, formal, one-shot interactions are the exception, rather than the rule, in interpersonal communications. The majority of office interactions consist of brief informal two-person exchanges, for example, to answer a colleague's question, to remind a coworker about a deadline, to hand over a document, or to discuss a social issue [Frohlich and O'Conaill 1995; Isaacs et al. 1997; Kraut et al. 1993; Suchman and Wynn 1984; Whittaker et al. 1994]. Here we therefore focus on brief, two-person, informal, repeated communications which we call *lightweight interactions.*

Research on scientific collaboration has shown that physical distance is the strongest predictor of whether two researchers will collaborate, precisely because physical proximity promotes lightweight interaction [Kraut et al. 1990; 1993]. The data we report below show exactly how a shared physical environment leads to these types of interaction. However, trends toward telework, mobile work, and the globalization of business are geographically separating workers, making collocated lightweight interaction less frequent and harder to achieve. In addition the use of technologies such as email and workflow may be decreasing the lightweight interaction that in the past took place around face-to-face document distribution and meeting scheduling [Whittaker 1996].

What is needed to support effective lightweight interaction at a distance? Technologies to support lightweight interaction will need to support its key features. Empirical work shows interpersonal communication[1] is (a) usually *two-person* rather than multiparty, so that for professional workers

[1]The data we report here cover *all* forms of interaction and not just lightweight interaction. They therefore include instances of more extended, formal, arranged meetings. By including all forms of interaction, these summary data thus *underestimate* some of the major characteristics of lightweight interaction—namely, their dyadic nature, brevity, opportunism, and repeated nature.

84% of meetings are dyadic [Whittaker et al. 1994]; (b) *brief,* with conversations generally lasting no more than a few minutes [Kraut et al. 1993; Whittaker et al. 1994]; (c) *opportunistic* rather than scheduled, with professionals having around 90% unscheduled meetings [Kraut et al. 1993; Whittaker et al. 1994], with figures of around 60% unscheduled meetings for managers [Panko 1992]; (d) focused around *shared objects* such as documents or designs [Luff et al. 1992; Tang 1991]; in our data we found that documents were involved in 53% of interactions [Whittaker et al. 1994]; and (e) *repeated* rather than one-shot. Repeated communications are often necessary because the purposes of lightweight interactions are seldom achieved in one interchange, with the result that such conversations are frequent and intermittent [Whittaker et al. 1994], with participants on average interacting with each other 2.5 times per day [Kraut et al. 1993].

To illustrate the character of lightweight interaction, we present two illustrative examples, taken from an extensive analysis of real workplace interactions. The data come from an observational study in which knowledge workers were "shadowed" for a week. We recorded all their interactions using a combination of video and audio. We generated a corpus of 294 conversations, involving a total of 99 different interactants. This research method and more detailed results are presented in Whittaker et al. [1994] and Frohlich [1995]. Both the current examples involve the same two participants, R and F, who work together in a surveyor's[2] office.

In Example 1.1, the entire two-person interaction is completed in four utterances and lasts only eight seconds. R sees that F is moving around the office and hence is not currently engaged directly in work. R therefore opportunistically solicits F's help. The fact that they share a common physical environment affords R this information about F's availability and allows F and R to jointly look at, and then physically exchange, the document. Note also the brevity of the interaction and the absence of formal conversational openings or closings, such as greetings or farewells.

Example 1.1. A short opportunistic interaction eliciting feedback about a document (duration: eight secs.).

R IS STANDING UP READING A DOCUMENT BEHIND HIS DESK WHEN HIS COLLEAGUE F WALKS INTO VIEW. F IS ON HIS WAY TO HIS OWN DESK FROM ANOTHER OFFICE.

1. R: "F, can you read this report for me?"
2. F: "Now?"
3. R: "Aye if you've got a minute."
4. F: "Yeah."

Example 1.2. An opportunistic interaction leading to unsolicited advice (28-second fragment of conversation lasting 1:36 mins.).

[2]In U.S. English, a surveyor is referred to as a real estate appraiser.

F IS ON THE PHONE ACROSS THE OFFICE FROM R. AS SOON AS F PUTS DOWN PHONE, R BEGINS TO SPEAK.

1. R: “Is he alright?”
2. F: “Yeah”
3. R: “Which one’s he’s got . . . there’s a restaurant”
4. F: “I said that I’ll do this one initially and then further afield”
5. R: “Which one’s that?”
6. F: “That’s 82 Whiteladies Road; it’s the offices”
7. R: “Oh, yeah we act for the landlord on that one. I did a rent review against him on that”
8. F: “Right”
9. R: “His shop it might be worth checking out. He’s got a subtenant downstairs who’s got a clothes shop”
10. F: “Yeah”
11. R: “Might be worth trying to get in with them as well”
12. F: “Yeah alright”

The fragment in Example 1.2 represents 28 seconds of a second interaction lasting 96 seconds. Although it again features R and F, it shows an unplanned conversation about a *different* topic, one that occurred immediately after F had finished a phone call to a client. It arose spontaneously because R heard F’s phone call and opportunistically wanted to monitor the outcome (line 1). Again the fact that participants share a common physical environment promotes this type of impromptu interaction. It continues with R offering unsolicited advice and assistance. Again the interaction starts without formal initiation, with R beginning with a direct question to F. The interaction also has a *history*, as revealed by the implicit shared context between the participants: without being told, R knows the identity of F’s caller (line 1) and details of the case (line 3), although R cannot remember all these details. R also proceeds to supply background information, which F may not already have known (line 7), and some advice (lines 9 and 11). The shared context results in a condensed and cryptic conversational style. This conversation continued for several more utterances after the extract: R gave more details and offered a warning about acting for both client and tenant. Thus an unplanned conversation led to a detailed task-oriented discussion initiated by R that was of benefit to F. Further analyses of these conversations, along with a demonstration of their functions and mutual benefits to the interactants, are given in Frohlich [1995] and O’Conaill and Frohlich [1995].

These results suggest a characterization of work in which people are engaged in multiple, intermittent, interleaved collaborative tasks. Workers seek out and are frequently sought out by their coworkers for *brief synchronous* opportunistic interactions. Each conversation may have a history of prior interactions, and workers are often concurrently engaged in multiple interaction threads. Furthermore, a worker may be engaged in multiple concurrent conversational threads with the same coworker. This gives rise to the problems of *context regeneration* and *tracking conversational threads*. Given that workers are engaged in multiple concurrent tasks, and that conversations are usually impromptu, how are workers able to switch context from their current task and immediately refocus on the topic that their coworker has just asked about? Furthermore, when the brief interaction is over, how do they switch back to their prior interrupted task [O'Conaill and Frohlich 1995]? How also do they keep track of the relationships between different fragments of the same intermittent conversation, when there are often long delays between related interactions?

We will argue that people currently exploit the presence of work-related artifacts (such as papers, drawings, notes, and folders) to help manage the history and context of these intermittent interactions. When task-related documents are copresent and visible, they can serve to "hold the context" of multiple ongoing conversations. Indirect support for this notion comes from our observation that 53% of workplace interactions involved a document [Whittaker et al. 1994]. Additional evidence for the context management function of documents is the frequently observed "messy desktop" [Barreau and Nardi 1995; Frohlich 1995; Kidd 1994; Malone 1984; Mander et al. 1992; Rouncefield et al. 1994]. Office workers scatter their physical desktops with clusters of notes, documents, and folders relating to ongoing projects. As a consequence, this information is readily at hand as a retrieval cue when an external interruption occurs. The "messy desktop" therefore allows people to regenerate the history of a prior conversation, even when substantial time has elapsed since the topic was last discussed.[3]

Other work has described people's attempts to use the computer desktop in an analogous way: electronic documents and folders are sometimes left visible in the user's electronic workspace to serve as reminders and context-holders for urgent work in progress [Barreau and Nardi 1995]. Similarly, users often retain undischarged email messages in their inboxes to serve as reminders about tasks or conversations that are still in progress [Whittaker and Sidner 1996]. Despite this, few systems provide direct support for context maintenance.[4] Furthermore, a number of recent studies have emphasized the utility of paper documents in providing an external visible record of current ongoing activity and have contrasted this with the relative inaccessibility of screen-based information for context tracking [Bowers 1994; Luff et al. 1992; Whittaker and Schwarz 1995].

[3]An alternative perspective is that the "messy" desktop is actually malfunctional, resulting from a lack of time or inclination to carry out filing and that it is an ineffective way to organize ongoing tasks. While we know of no study that has quantitatively compared the success of "messy" and "neat" desktops, there is strong evidence of the reminding functions of visible materials [Whittaker and Schwarz 1995; Whittaker and Sidner 1996], as well as the ability of users to find information in "messy" piles [Barreau and Nardi 1995].

[4]One exception is a system by Henderson and Card [1986] that uses the metaphor of "rooms" into which users can place related documents or applications, where it is easy to switch rooms and hence contexts.

These paper documents and notes also serve a second role. Once an interaction is underway they function as *shared objects.* They operate as part of a shared workspace and provide a shared physical context for the conversational participants. They also act as (1) a resource for looking up and recording information and (2) as a target for gesturing, marking, and note-taking [Frohlich 1995; Luff et al. 1992; Tang 1991; Whittaker et al. 1994]. Studies of remote synchronous communication using audio and shared workspace have also documented these functions for electronic documents and shared materials as "context-holders" and memory aids in real-time discussions [Bly 1988; Brinck and Gomez 1992; Minneman and Bly 1991; Whittaker 1995a; Whittaker et al. 1991; 1993]. Similar benefits occur when groups are provided with a shared *dynamic* image to support collaboration at a distance [Gaver et al. 1993; Nardi et al. 1993; 1996; Whittaker 1995a; 1995b]. In these applications video supports a shared workspace by providing joint access to video images of work objects that are critical to the collaborative task of the distributed group. In a neurosurgery application [Nardi et al. 1993; 1996; Whittaker 1995a; 1995b], remote members of a distributed surgical team were able to view video images from the operating theater of the patient's brain or spine. They were therefore able to see the state of the operation "at a glance" by looking at the image. This enabled them to coordinate their remote actions and provide advice to the team in the operating theater.

Using artifacts as reminders and context-holders for ongoing tasks is also important in the event of *failed attempts* to initiate lightweight interaction. We refer to this as the *connection problem.* The *opportunistic* nature of lightweight interaction means that attempts to initiate communication often fail, because the intended conversational participant is not currently available for conversation. Thus we found that attempts to initiate impromptu communications using the telephone failed on 62% of occasions, because the intended recipients were away from their desks or otherwise engaged in communicating with another person [Whittaker et al. 1994]. Similar failure rates are reported by other studies [Rice and Shook 1990]. This suggests two new communication requirements: *one-way drop* and *communication reminders.* For many communication purposes, one-way information transmission may be sufficient: surveys report that users feel that leaving a message is adequate for over half of business telephone conversations [Rice and Shook 1990]. For this reason, office workers often leave each other brief notes accompanying documents ("here's the most recent draft; can you look at pages 3–5?"). On other occasions, however, one-way drop of information may be insufficient. It may therefore be necessary to have synchronous communication, in which case a *communication reminder* may be important: leaving a message ("call me before 11, on 123 4567") can be used to coordinate a future synchronous connection [Tang et al. 1994]. Following from this, a further function of desktop notes, documents, and folders may also be to *remind* the worker of the fact that a connection attempt failed and that a synchronous conversation still needs to take place. Furthermore, artifacts can *hold context* during the period that one is waiting to communicate synchronously with another person. Thus when the connection is successfully made, the original conversational context can be more easily regenerated, because the presence of the documents serves as a retrieval cue for memory access.

This analysis of workplace communications indicates that effective support for lightweight interaction requires five related components:

(a) *Conversational threading:* Participants are engaged in multiple intermittent communications tasks, often with different individuals. The system must therefore keep track of interactions, storing elements of the same conversation together so that they can rapidly be accessed as a unit, allowing participants to check the status of a given conversation.
(b) *One-way drop:* The system should support the ability to leave a brief asynchronous message, given the fact that attempts to achieve opportunistic connections frequently fail, and a valuable-information exchange can often take place without synchronous communications.
(c) *Quick connection:* Given the brevity of lightweight interaction, the system should support rapid flexible communications, and participants should be able to quickly connect with others.
(d) *Context preservation and regeneration:* Given the intermittent but repeated nature of lightweight interaction, where there are often long delays between elements of the same conversation, the system should support methods for straightforwardly accessing prior parts of ongoing conversations, including the materials or artifacts that are involved in that interaction.
(e) *Shared objects:* The system should support real-time shared objects as props and conversational resources, both because of the frequency with which documents feature in lightweight interaction and their supporting role in mediating conversation.

Principles (a) and (d) and to a lesser extent (b) require the management of interactions across time, a problem which has characteristically been ignored in theoretical, empirical, and technology-oriented work [O'Conaill and Frohlich 1995; Whittaker and Sidner 1996; Whittaker et al. 1994]. In contrast, (c) and (e) are more concerned with real-time interaction, which has received more attention. Overall we need to support both synchronous and asynchronous communication methods, as part of an *integrated* set of communications applications for lightweight interaction [Tang et al. 1994]. Users can then choose the appropriate communication method as the situation demands.

2. CURRENT TECHNOLOGIES FOR SUPPORTING LIGHTWEIGHT COMMUNICATIONS: STRENGTHS AND WEAKNESSES

Our discussion has mainly focused on analyzing lightweight communications in workgroups who share the same physical location. We now evaluate current technologies used for remote communication to see how well they support the lightweight-interaction characteristics identified above.

Table I. Evaluating Current Communications Technologies for Lightweight-Interaction Features

Type	Telephone	Voicemail	Email	Workflow	Pager	Media Spaces	Videophone/ Videoconference	Shared Workspaces
Task Threading	No	No	Weak support in some systems	Yes	No	No	No	No
One-Way Drop	No	Yes	Yes	Yes	Yes	Not supported in most systems	No	No
Quick Synchronous Connect	Yes	No	No	No	No	Yes	Yes	Yes
Context Regeneration	No	No	Weak support in some systems	Yes	No	No	No	No
Real-Time Shared Objects	No	No	No	No	No	Not supported in most systems	Not supported in most systems	Yes

Table I shows that no current technology supports all the features of lightweight interaction. Technologies tend to focus on purely synchronous communications (e.g., telephone, shared workspace, videophone) without support for context regeneration or task threading. Alternatively, they are focused on asynchronous communication (e.g., email, workflow, and voicemail), which is highly effective for information drop-off, but does not support real-time exchanges. Furthermore, we will argue that with the possible exception of workflow, most asynchronous technologies do not support context regeneration and task threading effectively.

The *telephone* supports synchronous connection, but it has no document integration or asynchronous components. It therefore provides no support for managing the history of an intermittent extended conversation by offering features such as *context regeneration* and *task threading*. It also provides no capability for *real-time object sharing*. The telephone alone does not support *one-way drop* of information, although combined with an answering machine it does so.

Voicemail does not support *synchronous communications* or real-time *shared objects*. Voicemail supports *one-way drop* of information, often as a means to promote synchronous communications, e.g., "this is X; call me back on 123 4567." Although there is message storage in voicemail, most systems limit storage to small numbers of messages which have to be deleted on a frequent basis,[5] and no commercial systems provide users with

[5]Part of the problem here is that the size of audio files makes it costly to store large numbers of messages, which precludes the creation of message archives.

ways to manage messages according to *thread* [Rice and Shook 1990]. Again this means that the prior *context* of a voicemail message is often lost.

Email does not support *synchronous connection.* It also does not allow users to *share objects* in real time. Given its asynchronous nature, email supports *one-way drop* of information. Most email systems also provide minimal features for *context regeneration,* in that they allow users to save messages. However, the majority of systems lack features that explicitly facilitate context management, such as the automatic categorization of messages by conversational task. For example, one might wish to view originating messages and responses to those messages together to determine the conversational context for a response one is generating or to track the state of an ongoing conversational thread [Whittaker and Sidner 1996]. Most email systems provide folders, for the categorization of messages, and hence provide weak support for *context management,* but categorization requires the user to execute each action manually, rather than being an intrinsic system feature.[6]

Workflow systems do not allow *synchronous connection,* or *shared objects,* but their primary aim is to allow users to track the set of interactions that are associated with complex transactions occurring over extended periods of time. They therefore support aspects of *threading* and *context regeneration* [Abbott and Sarin 1994; Winograd and Flores 1986]. Workflow systems are asynchronous and in principle support *one-way drop,* although they tend not to be used this way, as they are intended to manage more extended interactions.

Pagers do not support any of the above features, being solely a means to *drop* information, such as a short message or phone number. They do not support any other communication features.

Media spaces are recently prototyped technologies that provide users with permanently open video/audio links or methods to quickly establish synchronous video/audio links [Bly et al. 1993; Fish et al. 1992; Gaver et al. 1992; Mantei et al. 1991; Tang et al. 1994]. Recent systems provide some integration with *real-time shared applications* to allow object sharing[7] [Tang et al. 1994]. Two systems [Gaver et al. 1992; Tang et al. 1994] also provide support for *one-way drop,* but none of these systems helps with *context regeneration* or *task threads.*

Videophones (and their multiparty equivalent, *videoconferencing*) are predominantly synchronous technologies, supporting *quick connections.* A few implementations have included *shared objects* [Tang and Isaacs 1993], but most do not. None of the other features such as *one-way drop, context regeneration,* or *threading* are supported by *videophones.*

---

[6]Email systems such as Notesmail™, Eudora™, and ccMail™ allow users to *program* automatic message classification, but some expertise is required to set this up.

[7]One can obviously "share" objects in a rudimentary manner by pointing a video camera at them [Gaver et al. 1993; Nardi et al. 1993; 1996; Whittaker 1996a; 1996b], but currently video does not provide high enough resolution to allow text to be easily read this way; nor does this method allow both partners equal ability to modify the document.

Shared workspaces support real-time *object sharing*, and a number of products exist, e.g., Proshare™, ShowMe™, and Deskslate [O'Conaill et al. 1994], although none is ubiquitous. These systems are intended to support *rapid connection*. There are no explicit features in these systems for *context regeneration, threading,* or *one-way drop* of information.

It is clear from the above that no current technologies support all five lightweight-interaction features. TeleNotes was designed to rectify this situation.

3. DESIGN

The observational data from face-to-face lightweight interaction enabled us to generate a set of five critical lightweight-interaction requirements to guide the design of TeleNotes. The aim was to present a unifying user interface metaphor for the applications that would support these five requirements.

3.1 Presentation Metaphor

The TeleNotes user interface is designed to be analogous to aspects of real-world paper-based interactions, to offer our users a familiar metaphor for system interaction. We present electronic equivalents of work-related objects such as documents, notes, and folders to offer office workers familiar tools with which to manage their lightweight interaction over time. TeleNotes is intended to resemble a real-world "messy desktop" containing papers which are laid out at specific spatial locations, with related information being arranged in stacks [Frohlich 1995; Kidd 1994; Malone 1983; Mander et al. 1992]. Information in TeleNotes is therefore spatially arranged around the computer desktop, with each stack being relevant to a separate ongoing lightweight-communication task. Each stack is automatically sorted so that it contains information relating to each ongoing conversation such as prior messages and relevant documents. These messages and documents are intended to serve as "context holders" for each separate intermittent interaction. As with their real-world equivalents, each stack can be physically arranged and relocated anywhere on the desktop by the user. Moving one item in the stack is sufficient to relocate the whole stack, given that TeleNotes automatically maintains the relations between stack items. The stacks "float" on top of other applications, making them highly visible.[8] This constant visibility means the stacks can remind the user about what communication tasks are currently in progress. The fact that stacks are readily accessible allows rapid context regeneration of materials related to a specific interaction by simply "opening" the stack of materials. The stacks are shown in Figures 1 and 2.[9] Figure 1 is a schematic of the computer desktops of two users Steve and

[8]Some of our users actually found the constant visibility too distracting, a problem we discuss later.

[9]Our figures are a mix of screen dumps and schematics; we use the latter for clarity of depiction.

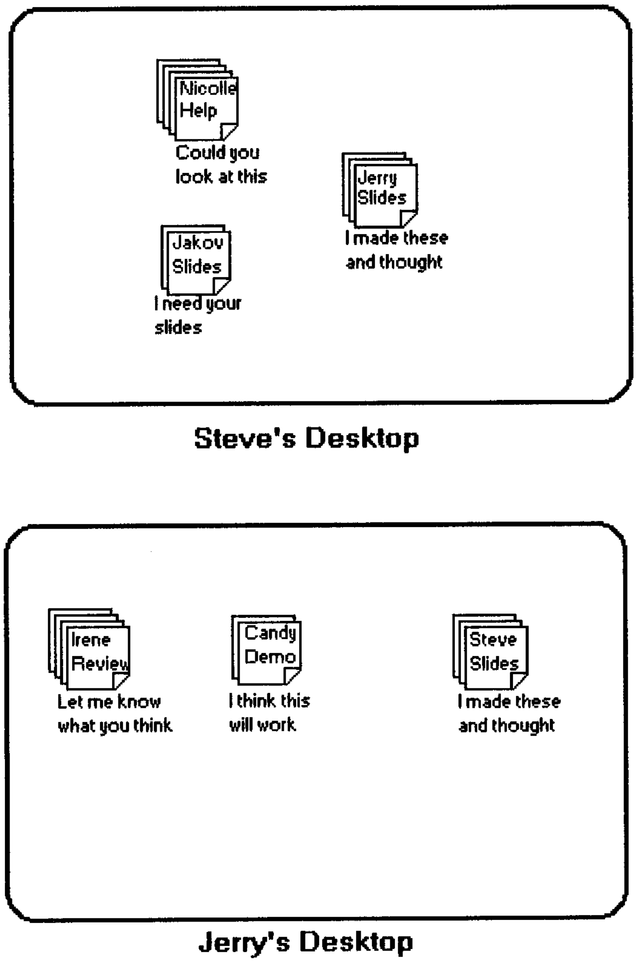

Fig. 1. Schematic showing management of three conversations using stacks, for two different users' desktops.

Jerry, who each have three ongoing communications represented by three different stacks. Figure 2 is a screen dump from TeleNotes showing a user's desktop with multiple stacks, as well as an open message which is part of one of the stacks (shown in the upper right-hand corner of the figure). We discuss the detailed features of the stacks later, when we examine *threading* and *context regeneration.*

How then does TeleNotes support communication? A second key notion is that of the "sticky."[10] Communications in TeleNotes are centered around

[10]A number of other recent programs have used the stickies metaphor, but in ways that are different from what we have done here. Apple Computer, Inc. devised a program in System 7.5™, which is intended to support a personal reminding function, e.g., allowing people to leave brief notes for themselves. It does not allow for communication with others however. Our application also contrasts with applications running under X Windows™, which allow users to

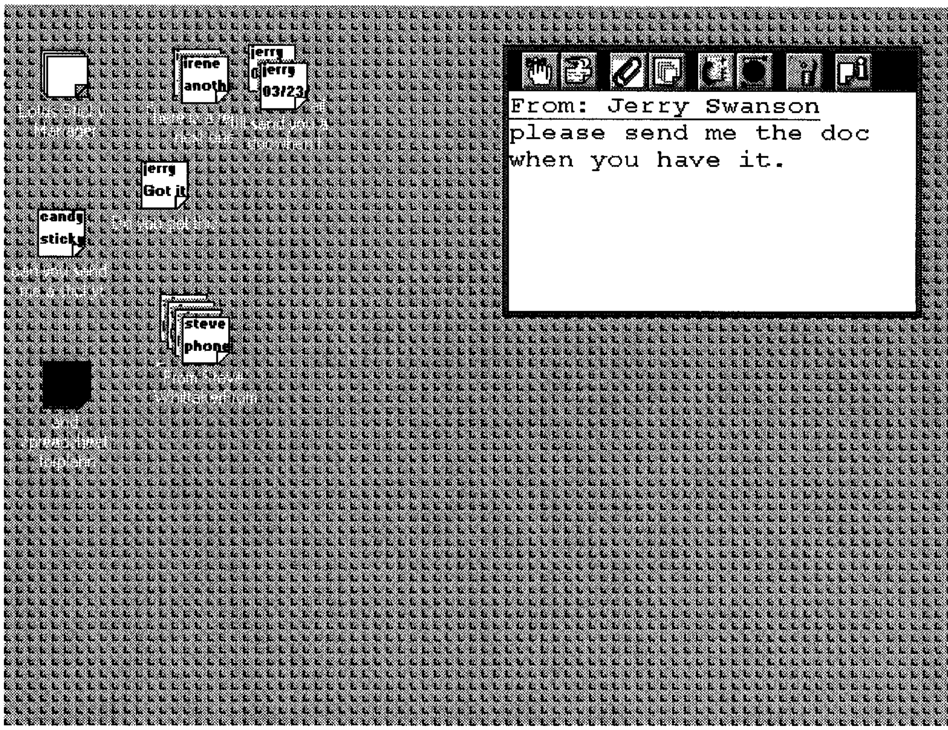

Fig. 2. Screen dump showing multiple stacks with one of the component communication messages of one thread shown "open."

"stickies" designed to be analogous with physical Post-its™. Stickies are small "document-like" objects, on which the user can type brief messages. Each communication stack or thread is constituted by a collection of sticky notes that all relate to a common conversational topic. The thread is therefore made up of a series of brief typed conversations about a related topic that are ordered in temporal sequence. When a sticky is sent to another user it appears on the remote desktop in a way that catches the recipient's attention, both because it appears on top of other applications and because it flashes periodically until it is read. If the sticky is part of a preexisting thread it will automatically appear at the same location as other stickies from that thread as the uppermost element on the stack (as depicted in Figure 1). TeleNotes does more than support message threads, however: sticky notes also serve as methods for both rapidly launching different types of synchronous communications and as ways of encapsulat-

send short messages. These applications do not support many of the features we developed here such as threading, attachments, synchronous audio/video communications, or object sharing. The most similar application is the "stickup" notes which are part of the Montage prototype [Tang et al. 1994]. Montage "stickups" support many of the communications functions described above, but not the *threading* or *context regeneration* functions addressed in TeleNotes.

ing or attaching documents and other applications. TeleNotes can be used to initiate synchronous voice and video communications and real-time object sharing, because stickies hold connection and addressing information about the sender. Whenever users send a sticky, information about their email, telephone, videophone, and network address is also implicitly encapsulated in the message, and this information can be used to make real-time connections. Finally, stickies serve as "cover notes," so that they can be attached to application files to allow for the attachment of materials such as documents or slides. Stickies are therefore containers for any class of material that is relevant to the ongoing interaction.

We now describe how TeleNotes addresses each of the five lightweight-interaction design requirements:

—*Conversational threading:* Participants are engaged in multiple intermittent communications tasks, often with different individuals. The system must therefore keep track of interactions, storing elements of the same conversation together so that they can rapidly be accessed as a unit, allowing participants to check the status of a given conversation.

Office workers are often engaged in multiple simultaneous projects about different topics, and threading can help them track each independent set of communication tasks. Figure 1 shows two users' separate desktops. Each separate conversation is represented in a different threaded stack appearing at a distinct location on the user's desktop. Whenever the user initiates a new sticky conversation (which is achieved by clicking on the StickyManager icon to generate a new sticky), a new stack or thread is created. Any response to that initial sticky will then be threaded onto the initial message, as will any subsequent message that is part of the same thread. We describe the technical details of how threading is implemented in the next section. Stacks are automatically labeled with the name of the communicating partners, and users can add their own task labels to a stack. TeleNotes also displays the first 20 characters of the uppermost sticky message under the stack.

Figure 1 shows that Jerry is currently working on three tasks (labeled Review, Demo, and Slides), each represented by a different stack. These are with Irene, Candy, and Steve, respectively. Steve is also working on three tasks, one of which (Slides) is also with Jerry. Thus if Irene asks Jerry a question about their ongoing conversation, Jerry will rapidly be able to identify the set of messages relevant to his interaction with her, along with their associated materials such as documents or slides. The stack metaphor therefore enables users to easily switch between different ongoing communications tasks, as the situation requires, and thus manage multiple intermittent conversations.

—*One-way drop:* The system should support the ability to leave a brief asynchronous message, given the fact that attempts to achieve opportunistic connections frequently fail and that valuable information exchange can often take place without synchronous communication.

Stickies provide a way to deliver an asynchronous quick message. Figure 2 shows Steve's desktop with an opened sticky, which is an asynchronous request from Jerry Swanson. New sticky threads are straightforward to create. They are initiated by the single simple action of clicking on the StickyManager (top left-hand corner of the screen), and brief messages can then be typed in. Addressing[11] and delivery are achieved through Notes Mail™. Stickies are delivered rapidly in a LAN environment. They appear on the recipient's desktop a few seconds after sending, threaded onto the relevant stack if they are part of an ongoing conversation. Furthermore, each sticky that is sent is automatically copied onto the sender's desktop, to serve as a record of that part of the ongoing conversation including the documents that are relevant to each point in the conversation. This copy is intended to *remind* users about current communication tasks: both *that* the task is in progress and about the content of the conversation.

—*Context preservation and regeneration:* Given the intermittent but repeated nature of lightweight interaction, where there are often long delays between elements of the same conversation, the system should support methods for straightforwardly accessing prior parts of an ongoing conversation, including the materials or artifacts that are involved in that interaction.

Users are often interrupted in the context of one conversational task and required to converse about another task. How does TeleNotes support this? Each stack is a record of prior elements of the conversation and is therefore a method of regenerating the context of the ongoing conversation. Figure 3 depicts the desktop of a user who has exchanged four sticky messages in the course of revising a document. Here the user has "opened" the stack (by a single menu selection) to show the history of one conversational thread. He can now access any prior part of the conversation with Jerry Swanson by consulting the relevant sticky.

Context is not only constituted by the prior sticky messages in a conversation; it also involves other types of documents and materials generated and used in the course of the conversational interaction. Stickies therefore also support the attachment of documents or applications, and they allow these documents or applications to be included in the conversation context. In this way stickies serve as "cover notes" or containers for the attached information. Figure 4 shows a conversation related to document editing, where the presence of an attached document is depicted by the underlining of the word "document" in the sticky, showing a hypertext link. By clicking on the link, the user can access the attached object, in this case the document shown in the lower part of Figure 4. Clicking on a link automatically launches the relevant application, which can be any desktop program. At this point the application is displayed only on the user's machine,

[11]The system uses the central Name and Address book in Notes Mail™ for addressing, so that the user only has to remember the name of the person for the sticky to be correctly addressed, and not their email address, username, or domain.

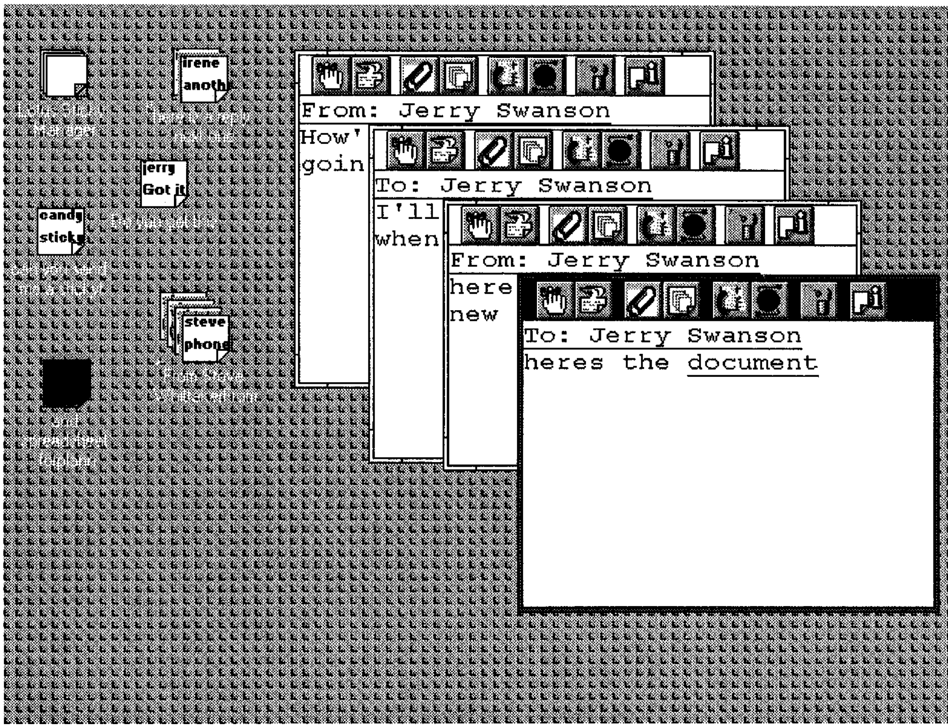

Fig. 3. An extended sticky conversation, showing conversational history.

but we describe below how two users can share the same object simultaneously using the communications options. The process of creating hypertext links is also straightforward: the user selects a menu option or icon and clicks on the relevant document. The hypertext link is placed under whatever text the user has highlighted or, failing this, the word which was nearest the position of the user's cursor when the menu option was selected.

Stickies with attached documents can also be used to support one method of document versioning. Thus, if users are unsure of the precise state of the document at a prior stage of an extended editing process involving multiple exchanges of documents, they can straightforwardly access prior versions of the document from the relevant prior stickies. They do this by identifying the relevant earlier stickies and clicking on the document link within each sticky. They can then make the relevant comparisons, with other later stickies and document versions. By using the stacked stickies as pointers to the attached documents, the user is therefore provided with a lightweight method of document versioning: by looking back through the stack the user can view the document at different stages of the revision process. This use of TeleNotes for versioning is schematically depicted in Figure 5.

We have so far described one aspect of the problem of retrieving the context of a prior conversation—namely, regenerating the information that is of immediate relevance to an interrupting coworker. There is also a complementary problem: after the interruption is over, people then have to

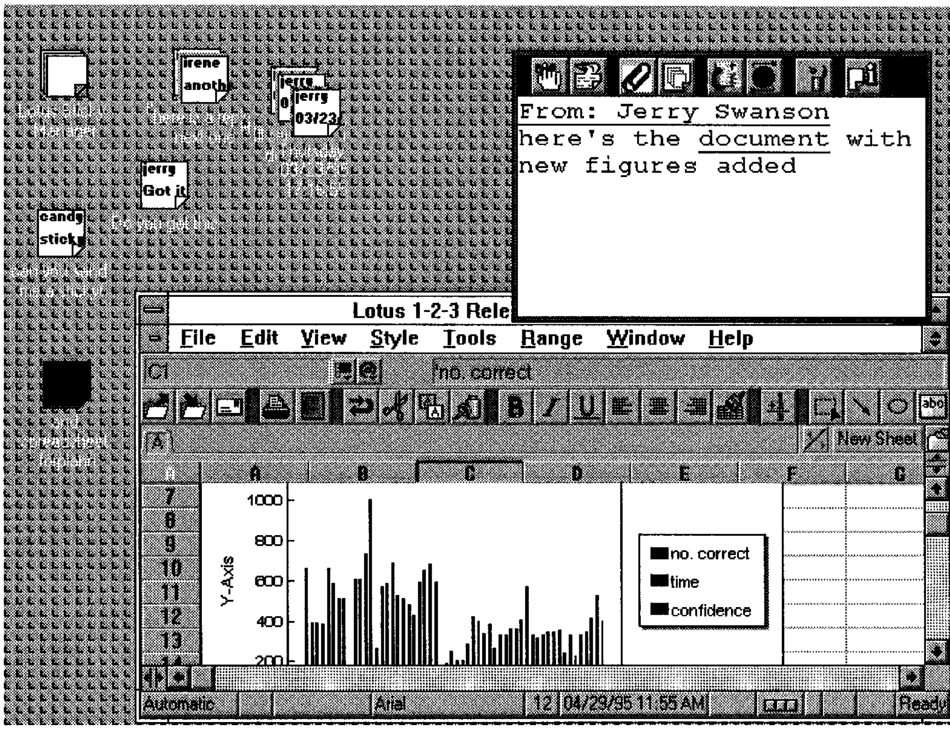

Fig. 4. Accessing a document from a hypertext link in a sticky.

regenerate the context of *their original task,* i.e., the one that the coworker interrupted. We have implemented a feature that enables users to save the context of current activity when interrupted. A user might therefore be compiling a document based on spreadsheet data, when a coworker interrupts asking a question about a different problem. By selecting the relevant menu option, the user creates a sticky which automatically incorporates hypertext links to all open applications and their most recently accessed files. By typing a quick description onto the sticky—"Q3 report and spreadsheet data"—the user can save the context of all the current applications, which can then be reinstated when the interruption is over.

—*Quick connection:* Given the brevity of lightweight interaction, the system should support rapid flexible communications, and participants should be able to quickly connect with others.

TeleNotes can also be used to initiate lightweight *synchronous* conversations. Stickies are a method of achieving rapid connection to other users, because stickies encapsulate connection and addressing information from the sender. Users can respond to a sticky by choosing one of a variety of communication media. Clicking on the sender's name[12] generates a series

[12]"Name" would be underlined, depicting a hypertext link.

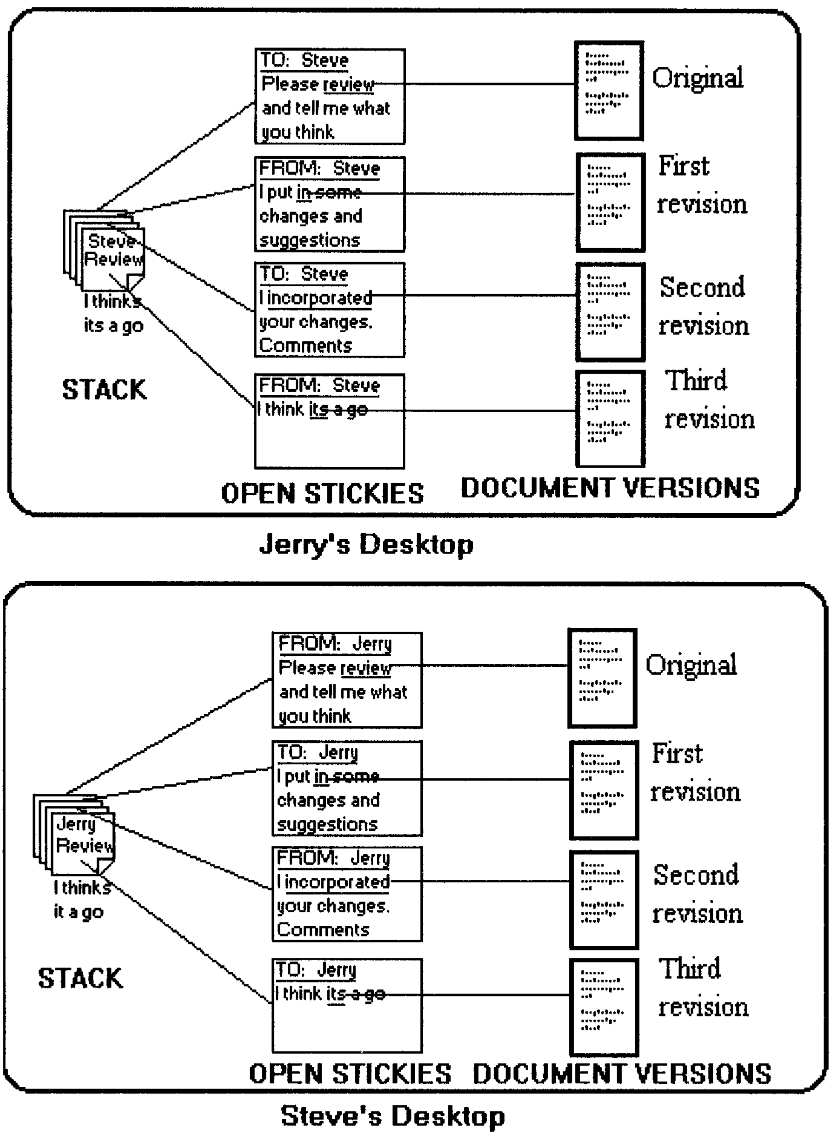

Fig. 5. Schematic showing how stickies can be used for versioning.

of connection options. These are shown schematically on the right-hand side of Figure 6 and are sticky, phone, videophone, email, and fax. On selecting the "phone" menu option, TeleNotes automatically dials the sender's telephone number and establishes a real-time audio link. In this sense, a sticky operates like a multimedia "business card," because it contains all the addressing information necessary to connect back to the sender. Note also that, because the connection information is embedded in the original sticky, this removes from the user the cognitive overhead of finding relevant addressing information and manually initiating connec-

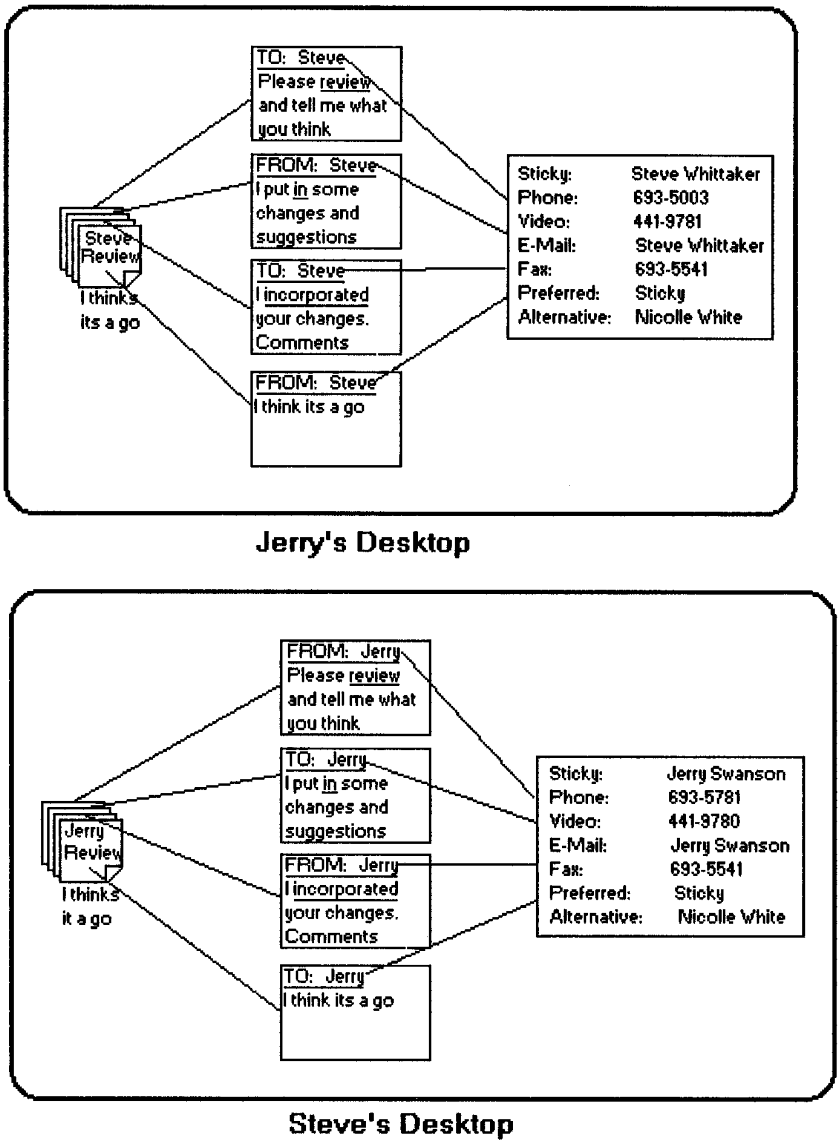

Fig. 6. An extended sticky conversation between two users showing connection alternatives accessible from a sticky.

tions, by typing or dialing numbers. The result is that connection is fast[13] and lightweight, addressing one of our initial design goals. We now discuss shared objects.

---

[13]This was true unless the video option was chosen, when videolinks often took 11 seconds or more to establish, given the limitations of the ProShare conferencing protocol and the restricted ISDN bandwidth it used.

—*Shared objects:* Given the document-focused nature of lightweight interaction, the system should support real-time shared objects as props and conversational resources, because of the frequency that documents feature in lightweight interaction and their supporting role in mediating conversation.

If users wish to synchronously share or edit a document (or any other application) they can do so in TeleNotes. Sharing is just another communication option depicted in Figure 6. For those users who have video hardware, TeleNotes will open an audio and video connection to the recipient. Once the connection is open the user can include the application, using the dedicated data-sharing program. For users without video, data sharing is accompanied by an audio link only. Once the channel is established, then object sharing can take place, again using the data-sharing program. Again because the originating sticky holds recipient addressing information, the user does not have to remember such information, so that selecting the relevant menu option automatically makes the connection. Again this means that object sharing is a lightweight and rapid process.

4. IMPLEMENTATION

The prototype uses the metaphors of stacks and sticky notes to support the multiple facets of lightweight interaction. In order to assure the prototype could be used by users in their everyday work, we based the implementation on existing software infrastructure. TeleNotes therefore relies on Notes Mail™ for communication transport and storage and on ProShare™ for audio/video and real-time data-sharing connections. Audio-only connection is achieved using CallCom™ software to switch ROLM dataphones. The novel components of TeleNotes are (a) the UI and the hypertext editor for the stickies themselves, (b) the Sticky Manager which handles threads and desktop presentation of related stickies, and (c) the transport protocol for the stickies.

The TeleNotes application runs in a networked Microsoft Windows™ environment (version 3.1). The architecture of the application involves two executables, StickiesManager and StickyInstance, and a shared DLL (dynamic linked library) CommonTransport. The application was developed in C++ using Microsoft Foundation Classes™ and the Microsoft Visual C++ Environment. The CommonTransport DLL uses the Lotus Notes™ API.

4.1 StickiesManager and StickyInstance

The StickiesManager executable manages each StickyInstance executable, the threading and presentation of StickyInstance task stacks, and the communication between all StickyInstances. The StickyInstance executable provides a text editor for composing stickies and manages the hypertext links, including the methods to attach and launch application files.

When a task or stack of stickies is begun, the StickyInstance registers with StickiesManager which in turn allocates each new StickyInstance by a Unique Task ID (UTI). Subsequent StickyInstances related to this stack or task (i.e., responses to the originating sticky) have the same UTI. The UTI is used by the StickiesManager to manipulate and track related StickyInstances. Any sticky that is generated by responding to a prior sticky automatically inherits the original sticky's UTI. The StickyManager uses this information in the spatial presentation of stacks and the set of manipulations that can be globally applied to a stack (e.g., "opening" all stickies or dragging a stack to a new location). The StickiesManager also stores the connection information associated with each stack, such as the sender's phone or ISDN connection information.

4.2 The Transport Process

When user1 sends a message to user2, user1's StickyInstance encodes the message (as well as hypertext attachments if there are any) and hands the encoded message to the StickiesManager. The StickiesManager embeds certain state and user connection information and the UTI. The CommonTransport DLL provides for the transport of StickyInstance text and attached documents using Lotus Notes Mail as the transport mechanism. The message is passed to the CommonTransport DLL which adds a Sticky signature (or key) and then uses Notes Mail to send the message to user2's Notes Mail database. The message is transmitted as compressed ASCII.

User2's StickiesManager polls its Notes email through the CommonTransport DLL protocol, looking for new mail messages with a Sticky signature. The StickiesManager moves these messages to a new Notes database and creates a new StickyInstance. The StickyInstance decodes the message and presents it to the user. When an incoming message is part of an existing thread, the StickiesManager detects this and presents it at the relevant spatial location, based on the location of the relevant stack. The Notes database used for the persistent storage of stickies can be either local or on a Notes server. In either case, users can work while not directly connected to a server. In addition, TeleNotes can be accessed and run over modems and analog telephone lines, because of the use of the underlying Notes transport.

4.3 Audio/Video and Data-Sharing Connections

The audio/video connection between Sticky users is accomplished using Intel ProShare™ Personal Conferencing Video System 200 (version 1.8), connecting over ISDN lines. For audio/video connections, the Sticky application uses Microsoft DDE to communicate with the ProShare application. The data-sharing conferencing uses Intel ProShare Premier edition software. In either case, the Sticky application communicates with the ProShare software by various Windows messages to pass relevant information about which connection options (audio, video, shared data) are requested along with (a) addressing information and (b) attached application information. When users request audio-only communication, we use a separate phone application. Here phone control is achieved using DDE to pass requests for phone connection along with addressing information to the ROLM Callcom™ application and then to the ROLM dataphone.

5. OBSERVATIONS AND USER FEEDBACK

A number of different versions of the prototype have been deployed and used by eight people in the course of their everyday work over a period of two months. The users consisted of researchers, administrative assistants, managers, and software engineers. Feedback and usage patterns were gathered by informal user interviews and comments elicited by email.

5.1 How Was It Used?

The informal field trial showed that the prototype was used for the types of brief communications that we had originally intended. Typical sticky messages were "Here's a quick draft of proposal X, comments please,"[14] "I left the hardware you need on your office chair," and "the files are on the server at location Y." Messages were short: few were more than two sentences long. These generated equivalently brief replies. Stacks were predominantly short, consisting of no more than three messages. Senders also frequently exploited the fact that TeleNotes enabled them to include documents as part of the message context.

Comments from users showed that for certain classes of communication, the stickies were preferred to email. They were described by users as "quicker to start up." Also because TeleNotes achieved fast message delivery, and stickies appeared almost instantaneously on the other person's desktop, it led to more "quickfire" exchanges than with email communication. As a result, people commented that they had "more of a sense" about what a collaborator was doing than when using email.

People also exploited the spatial aspects of stacks in TeleNotes, to engage in task management. They placed different communication task threads at different desktop locations to manage different ongoing communications. Users seldom maintained more than four different stacks at any one time (for reasons we return to later). In addition, TeleNotes was used for *personal* reminding, with people using it for sending "messages to themselves," e.g., "remember to confirm flight," relying on the visibility of stickies to guarantee they saw the message. Again these reminders were highly brief, consisting of a few words only. One unforeseen usage was to display large untransmitted stickies on one's *own* machine to communicate information to others. Thus one user left a sticky on her computer desktop saying "I'm out of the office today, contact phone, 123 4567," so that others could see it when they came into her office. The advantage she stated was that she could create this message from offsite and still have it displayed in her office, where people would expect to find her.

[14]This had a document attached to the message, i.e., the hyperlink to "draft."

5.2 Design Feedback

The functionality of the application was apparent to users, and the UI presented them with few difficulties, so that people were able to use it after a demonstration lasting only a few minutes. Users made a number of comments about our design.

People said that on occasion the threading mechanism could be confusing, especially when there were several ongoing communications tasks. This may explain why users generally restricted themselves to four or fewer threads. Confusion was most pronounced when people were engaged in multiple different threads with the same person. Although we provided the facility for users to specify task labels which should have differentiated the different threads, there were occasions when users gave tasks similar labels. One solution here might be to "color-code" threads from the same user to reduce the chance of confusion. Two users observed that one of their threads became sufficiently complex that they wished to generate subthreads. In the current TeleNotes model, however, this is not possible, although the design could be modified to incorporate it. Another problem was caused by a user generating two successive stickies without waiting for a response. Few users exploited the context preservation feature that enabled them to save all current objects and applications inside a sticky to preserve current work when being interrupted. It required them to select a special class of sticky and then type a descriptive label. This was perceived to be too cumbersome by users whose main priority on being interrupted was to address the interruption itself, rather than saving their current work.

Several users observed that there were restricted sets of people to whom they would send a sticky, i.e., that the acceptability of this communication method depends on the relation between the interactants. Our users felt that stickies were appropriate for peer-to-peer communication, but did not want to send stickies to strangers or high-level management. This may be indicative of the fact that we had succeeded in our objective of supporting informal rather than formal communications, as stranger- or status-oriented communications are characterized by their formality. A related issue concerned the "alerting" function when a sticky arrives, as signaled by a flashing desktop icon, overlaying the application that the user is currently working in. Users such as administrative assistants commented on the utility of this feature in enabling them to be responsive in their jobs. They noted the advantages of stickies over email in giving them immediate visual access to urgent messages. They pointed out that the arrival of the sticky could be detected, whatever application the person was using, and when they were on the phone. In contrast, urgent email messages may not be read for some time if the user is not currently using the email application. However, other users did not like the intrusiveness of the alerting function. They commented that it distracted them from their current activity. Together these results suggest that the utility of the application and aspects of its design may have to be sensitive both to the job type of the users and the relationship between the interactants. The design implication is that certain features such as alerting need to be user configurable. In addition, we need more research to determine which sets of users and workgroups most require this form of informal quickfire communication.

A different type of problem arose indirectly from the fact that TeleNotes is a communications application. With communications applications, it is necessary to design for the possibility that not all users are guaranteed to be running the application: one user might therefore send a sticky, but if the recipient is not running the application, it is necessary to guarantee that the recipient received the message. In our first prototype, we did not design for this eventuality. A redesigned version of TeleNotes provided a mechanism whereby sticky messages were delivered into email, if the application was not running. Email delivery obviously compromised the desktop delivery and "urgency" aspects of the application, but meant at least that messages were not lost altogether. One unexpected problem arising from this is that some users wanted to recompose their message if they knew it would be delivered in email rather than as a sticky.

Similar problems arising from the nature of communications applications were encountered with different hardware configurations. Not all users had the installed hardware to support all the communications options, e.g., application sharing and videophone. The fact that the hardware was not ubiquitous meant that our evaluation data are restricted for both audio, video, and application-sharing features. Again, this is independent of the design of TeleNotes, but nevertheless has an impact on whether and how the application was used. To this end, we constructed a version of TeleNotes which supported communication via stickies and email only, so that users could participate in TeleNotes communications, but without multimedia and application-sharing capabilities.

The available data on the connection options show that the audio connection was perceived as being valuable. Audio connection to the sender of a sticky was almost instantaneous, which was seen as being highly useful in allowing people to ask a quick synchronous verbal question about a sticky or attached document they had received. Connection time was much longer for video and data sharing however. Given that connection sometimes took as long as 11 seconds, video and data-sharing connections were considered to be too slow when the resulting conversations were intended to be brief.

People also expressed concerns about screen real estate. Stickies are deliberately designed to persist and remain visible on the user's desktop, as "reminders" about communication tasks in progress, and people stated that this was a useful feature. A difficulty, however, is that stickies also consume space on the desktop: by the time several conversational threads are underway, the space for viewing other desktop applications is much reduced. We therefore provided a method for users to "save" threads into a Notes database, thus removing them from the desktop. We also provided users with two presentation modes of running TeleNotes, in which stickies could “float” on top of or below the current application. The problem with putting stickies in databases, or having them submerged beneath applications, however, is that they reduce the visibility of these materials and compromise reminding and rapid context regeneration functions. For this reason, the visibility of stickies needs to be user configurable.

6. CONCLUSIONS

We built a system to support key aspects of lightweight interaction. The system addresses some of the limitations of other communications technologies in supporting *connection, context management, conversational tracking, shared objects,* and *one-way information drop.* Our early prototypes were successfully used by people in the execution of their everyday work. Nevertheless, there are still a series of intriguing questions concerning possible extensions to TeleNotes. These should be premised on more theory and data about lightweight interactions, which should help better focus future design choices. This section first compares our results with related systems built to support lightweight interaction. We outline existing theories of lightweight interaction, suggesting where questions remain, and then discuss the implications of these theories for future lightweight-interaction system design.

6.1 Related Systems

A number of systems have been built that partially address aspects of lightweight interaction [Bly et al. 1993; Fish et al. 1992; Gaver et al. 1992; Harrison et al. 1994; Mantei et al. 1991; Tang et al. 1994]. A key difference between the approach taken by all these other systems and the one taken here is that a major focus of our design was to support the construction and maintenance of shared context across repeated but intermittent interactions. To this end, we explicitly constructed tools to support the *prior history* of the interaction and the set of *shared objects generated in prior interactions,* as well as to allow real-time interactions to take place. The threading features and real-time communication capabilities in TeleNotes were also intended to address some of the perceived problems with asynchronous technologies like email, such as lack of explicit support for managing conversational history and poor responsiveness [Whittaker and Sidner 1996].

Another issue concerns the relationship between the lightweight interactions we supported here and systems built to support *awareness.* Awareness arises from implicit information about the actions of coworkers, and its function is often to help determine their availability for interaction [Bly et al. 1993; Dourish and Bly 1992]. While we did not attempt to explicitly support awareness in TeleNotes, our users’ comments indicated a heightened sense of the activities of coworkers. This may have arisen from the simple fact that stickies tended to be exchanged very frequently, giving them more rapid updates and more timely information about the current actions of colleagues.

6.2 Theories of Communication

We need to extend our understanding of lightweight interaction, both by building better theories and collecting more data. With certain exceptions [Isaacs 1989], current conversational theories have so far focused on *one-shot* interactions, having identifiable beginnings and endings, with conversational context evolving *throughout* the discussion [Clark and Brennan 1991; Clark and Schaefer 1988; Grosz and Sidner 1986; Sacks et al. 1974]. In contrast, our observational data showed that lightweight interactions often seem to consist of *multiple, distinct* unplanned fragments. Context has to be reconstructed on the *reinstigation* of an ongoing conversation after long periods have elapsed or after other activities have intervened. Conversational theories therefore need to be extended to address these new problems of *connection, context regeneration,* and *conversational tracking* to determine how people successfully initiate conversations, how they reinstantiate context when there are long intervals between interactions, and how they keep separate the multiple simultaneous workplace conversations they engage in.

Such theory needs to be specific about what variables affect these processes. In our initial observational study [Whittaker et al. 1994], and in testing our prototype, we found that *connection* and *initiation* were affected by the *relationship* between the communicating participants and by their job type. Thus administrative assistants felt that having arriving sticky notes "flash" on the desktop was crucial, because they did not wish to miss incoming urgent information. In contrast, other users found this type of alerting intrusive and disruptive of their current work. Other research also indicates that connection is highly sensitive to the relation between participants [Fish et al. 1992; Tang et al. 1994]. In addition, the *medium* of initiation can influence the initiation process; for example, video initiations may be radically different from those happening face-to-face [Heath and Luff 1991]. Having clearer principles about the factors that influence initiation will allow us to determine how to implement the necessary different styles of *connection.*

We also need to better understand what affects *context regeneration* and *conversational tracking*. Thus, the *number of simultaneous conversations* that the user is trying to track may influence these. The more conversations a person is engaged in, the harder it may be to effectively switch among them. The fact that our users in TeleNotes tried to restrict the number of current threads is also of relevance here. In addition, the *temporal pattern of interactions* may also be crucial: longer intervals between successive elements of a given interaction may make it more difficult to regenerate context. Another parameter is the *complexity* of each conversational task: so that tasks involving highly complex sets of materials may therefore be much harder to retain and remember than simple ones [Whittaker et al. 1993]. Our observations of usage were consistent with this, showing that stickies were often used for attachments of complex materials such as documents, which helped to manage the context of complex interactions.

We also need to know more about the processes by which people regenerate context. Research from discourse theory may be relevant here. Do people rely on linguistic cues to regenerate context [Grosz and Sidner 1986; Isaacs 1989; Litman and Hirschberg 1987; Reichman 1985; Walker 1992; 1993; 1996; Walker and Whittaker 1990; Whittaker and Stenton 1988]? We have argued here that people also rely on the presence of relevant physical materials for context regeneration, although this claim needs more systematic quantitative evaluation [Whittaker and Schwarz 1995; Whittaker and Sidner 1996; Whittaker et al. 1994]. There are other outstanding problems with regard to conversation management [Isaacs 1989]. How do people distinguish among different conversations especially when they are engaged in multiple interactions with a single person? How do people manage interruptions? On being interrupted, do they ever intentionally prepare for context regeneration by leaving themselves reminders for when they return to a task, i.e., how do they prepare for future retrieval?

Current communication theories have also tended to focus almost exclusively on synchronous communications, although we have seen here both that there are *connection* problems in trying to establish such communications opportunistically and that *one-way drop* of information can be highly effective in the event of failed attempts to connect [Rice and Shook 1990]. There is some theoretical work showing the importance of synchronous communication in tasks that require ambiguity resolution and consensus, such as planning or negotiation [Finholt et al. 1990; Kraut et al. 1990]. However, more research is required to clarify the respective benefits of synchronous and asynchronous communication and to determine how users make choices between these communication methods, when both are available.

6.3 System Extensions

Answers to the above questions will be invaluable in making design decisions about a set of possible extensions to our prototype. More detailed information on *connection* should help us to determine how best to implement the alerting function when a sticky arrives on a person's desktop. It should also determine whether we need to add the ability to start a conversation by visual "glancing" at a potential recipient using video, a technique which has been suggested as one solution to the connection problem [Bly et al. 1993; Fish et al. 1993; Gaver et al. 1992; Tang et al. 1994]. Alternatively, "open channels" [Bly et al. 1993] or "awareness servers" [Dourish and Bly 1992] may partially address the connection problem, by providing continuous updates about coworkers' activities. Clearer understanding of conversation tracking and context should also inform us about the utility of some of the novel features that we are currently designing, which are described below. We also need to know how *fast* connections need to be made. Our observations are consistent with other research in finding that delays of several seconds in achieving connection can compromise the utility of this type of application for brief communications [Tang et al. 1994]. This requirement for rapid connection presents considerable technical problems for wide-area applications.

The novel features we are exploring include the idea of *automatically* supporting aspects of context regeneration, to provide direct access to relevant prior contexts when a user experiences an external interruption, such as a telephone call. One possibility is to exploit information about the *identity* of the interrupter to trigger relevant contextual information. Incoming videophone, telephone, or network "caller-ID" information could be used by the system to automatically access relevant conversational threads. The current TeleNotes implementation has a database entry associating caller telephone numbers with their names (see Figure 6). Thus, individual stickies or whole threads associated with an incoming caller could be triggered and automatically accessed and presented to the user, making relevant documents and messages from prior interactions immediately available.

On other occasions, the user might want to be reminded of a particular context at a *specific time.* This may be applicable for a series of organized meetings, where a user might wish to retrieve the minutes of the prior meeting in the series, immediately prior to going into the next meeting. One technique might be to program individual stickies, so that context regeneration could be triggered by TeleNotes, at the time it is relevant for the user to have that information. One might therefore request a sticky to "reappear on Tuesday 7th at 9:45 a.m." to give the user 15 minutes to peruse the relevant contextual information immediately prior to the 10:00 a.m. meeting. Programmable stickies could also be used to promote connection: thus one user might send another user a sticky saying "call me at 4:00," which if accepted by the recipient would execute the relevant action. Programmable stickies might also address part of the problem of screen real estate. As our users commented, the more sticky stacks that one has concurrently displayed on one's computer desktop the less effective they become as visual reminders. Programmed stickies can be designed to only appear when the user has determined it is relevant for them to be visible, and hence reduce the total number of visible stacks.

In addition to more theory and evaluations of possible new design extensions, we need to understand the *role* of the lightweight interactions more precisely. We need to determine *the classes of conversation* for which the different aspects of the tool are most suitable. Thus our users' comments indicated that there were only certain types of situations in which they would use the stickies component of TeleNotes. A clear factor here was the perceived *informality* of stickies. Various people stated they would not use stickies to communicate with strangers or high-level management, i.e., for more formal types of interaction. This suggests a role for stickies within distributed workgroups who already know each other. In more-formal settings the other communications options (phone, fax, email) may have to be used, given the perceived informality of stickies. Other necessary requirements for the effective deployment of the whole TeleNotes system are that the workgroup's interactions are *repeated but intermittent.* TeleNotes offers fewer benefits if the conversation consists of a single interaction. Furthermore, if the conversational materials are simple and straightforward, then context regeneration may be unproblematic, so that TeleNotes may again offer reduced benefits, over phone or email.

Finally the design of TeleNotes reflects a recent trend in systems that attempt to exploit some of the features of paper-based systems, and their associated work practices, by incorporating these features into electronic applications. Recent empirical work has emphasized the benefits of paper, drawing attention to its visible nature, the fact that it can be easily spatially organized, and its context-holding function [Bentley et al. 1992; Bowers 1994; Luff et al. 1992; Whittaker and Schwarz 1995]. Systems have also been designed that attempt to provide synergy between paper and electronic worlds, e.g., "augmented reality" [Johnson et al. 1993; Mackay and Pagnini 1994; Newman and Wellner 1992]. Other electronic systems have explicitly been designed to model aspects of the paper world [Levine and Ehrlich 1991; O'Conaill et al. 1994; Whittaker et al. 1993; Wolf et al. 1989]. We need more work investigating the impact of some of the features of the paper world investigated here, such as visibility and ease of arrangement. Future electronic systems can therefore better exploit the affordances of the world of paper, including its utility in managing extended intermittent interactions that we investigated here.

ACKNOWLEDGMENTS

Thanks to Irene Greif, Sal Mazotta, John Patterson, Lyn Walker, Nicole White, and Brid O'Conaill for conversations about this work. Thanks also to the people who participated in the field trials.

REFERENCES

ABBOTT, K. AND SARIN, S. 1994. Experiences with workflow management: Issues for the next generation. In *Proceedings of Conference on Computer Supported Cooperative Work.* ACM Press, New York, 113–120.

BARREAU, D. AND NARDI, B. 1995. Finding and retrieving information: File organization from the desktop. *SIGCHI Bull. 29,* 3, 39–45.

BENTLEY, R., RODDEN, T., SAWYER, P., SOMMERVILLE, I., HUGHES, J., RANDALL, D., AND SHAPIRO, D. 1994. Ethnographically informed systems design for air traffic control. In *Proceedings of Conference on Computer Supported Cooperative Work.* ACM Press, New York, 287–298.

BLY, S., HARRISON, S., AND IRWIN, S. 1993. Media spaces: Bringing people together in a video, audio and computing environment. *Commun. ACM 36,* 1, 28–45.

BOWERS, J. 1994. The work to make a network work. In *Proceedings of Conference on Computer Supported Cooperative Work.* ACM Press, New York, 287–298.

BRINCK, T. AND GOMEZ, L. 1992. A collaborative medium for the support of conversational props. In *Proceedings of Conference on Computer Supported Cooperative Work.* ACM Press, New York, 171–178.

CLARK, H. AND BRENNAN, S. 1991. Grounding in communication. In *Perspectives on Socially Shared Cognition,* L. B. Resnick, J. Levine, and S. Teasley, Eds. APA Press, Washington, D.C.

CLARK, H. AND SCHAEFER, E. 1989. Contributing to discourse. *Cog. Sci. 13,* 3, 259–292.

DOURISH, P. AND BLY, S. 1992. Portholes: Supporting awareness in a distributed workgroup. In *Proceedings of CHI'92 Human Factors in Computing Systems*. ACM Press, New York, 541–547.

EGIDO, C. 1988. Video conferencing as a technology to support group work: A review of its failures. In *Proceedings of Conference on Computer Supported Cooperative Work*. ACM Press, New York, 13–24.

EGIDO, C. 1990. Teleconferencing as a technology to support co-operative work: A review of its failures. In *Intellectual Teamwork,* J. Galegher, R. Kraut, and C. Egido, Eds. Lawrence Erlbaum, Hillsdale, N.J.

FINHOLT, T., SPROULL, L., AND KIESLER, S. 1990. Communication and performance in ad-hoc task groups. In *Intellectual Teamwork,* J. Galegher, R. Kraut, and C. Egido, Eds. Lawrence Erlbaum, Hillsdale, N.J.

FISH, R., KRAUT, R., ROOT, R., AND RICE, R. 1992. Evaluating video as a technology for informal communication. In *Proceedings of CHI'92 Human Factors in Computing Systems*. ACM Press, New York, 37–48.

FISH, R., KRAUT, R., ROOT, R., AND RICE, R. 1993. Video as a technology for informal communication. *Commun. ACM 36,* 1, 48–61.

FROHLICH, D. 1995. Requirements for interpersonal information management. In *Mobile Personal Communications and Co-operative Working*, P. J. Thomas, Ed. Alfred Waller and Unicom Seminars.

GAVER, W., MORAN, T., MACLEAN, A., LOVSTRAND, L., DOURISH, P., CARTER, K., AND BUXTON, W. 1992. Realizing a video environment: EuroParc's RAVE system. In *Proceedings of CHI'92 Human Factors in Computing Systems*. ACM Press, New York, 27–35.

GAVER, W., SELLEN, A., HEATH, C., AND LUFF, P. 1993. One is not enough: Multiple views in a media space. In *Proceedings of CHI'94 Human Factors in Computing Systems*. ACM Press, New York, 335–341.

GROSZ, B. AND SIDNER, C. 1986. Attentions, intentions and the structure of discourse. *Comput. Ling. 12,* 175–204.

HARRISON, B., MANTEI, M., BEIRNE, G., AND NARINE, T. 1994. Communicating about communicating: Cross disciplinary design of a media space interface. In *Proceedings of CHI'94 Human Factors in Computing Systems*. ACM Press, New York, 124–130.

HEATH, C. AND LUFF, P. 1991. Disembodied conduct: Communication through video in a multimedia environment. In *Proceedings of CHI'91 Human Factors in Computing Systems*. ACM Press, New York, 99–103.

HENDERSON, D. AND CARD, S. 1986. Rooms: The use of multiple virtual workspaces to reduce space contention in a window based graphical user interface. *ACM Trans. Graph. 5,* 211–243.

ISAACS, E. 1989. Mutual memory for conversation. Ph.D. thesis, Stanford Univ., Stanford, Calif.

ISAACS, E., TANG, J., AND MORRIS, T. 1996. Piazza: A desktop environment supporting serendipitous and planned interactions. In *Proceedings of the Conference on Computer Supported Cooperative Work*. ACM Press, New York, 315–324.

ISAACS, E., WHITTAKER, S., FROHLICH, D., AND O'CONAILL, B. 1997. Information communication re-examined: New functions of video in supporting opportunistic encounters. In *Video-Mediated Communication,* K. Finn, A. Sellen, and S. Wilbur, Eds. Lawrence Erlbaum Press, Hillsdale, N.J.

JOHANSEN, R. 1984. *Teleconferencing and Beyond*. McGraw-Hill, New York.

JOHNSON, W., CARD, S., JELLINEK, H., KLOTZ, L., AND RAO, R. 1993. Bridging the paper and electronic worlds: The paper user interface. In *Proceedings of CHI'93 Human Factors in Computing Systems*. ACM Press, New York, 507–512.

KIDD, A. 1994. The marks are on the knowledge worker. In *Proceedings of CHI'94 Human Factors in Computing Systems*. ACM Press, New York, 186–191.

KRAUT, R. AND STREETER, L. 1996. Co-ordination in software development. *Commun. ACM 38,* 1, 69–81.

KRAUT, R., EGIDO, C., AND GALEGHER, J. 1990. Patterns of communication in scientific research collaboration. In *Intellectual Teamwork,* J. Galegher, R. Kraut, and C. Egido, Eds. Lawrence Erlbaum Press, Hillsdale, N.J.

KRAUT, R., FISH, R., ROOT, B., AND CHALFONTE, B. 1993. Informal communication in organizations. In *Groupware and Computer Supported Co-operative Work,* R. Baecker, Ed. Morgan Kaufman, San Mateo, Calif.

LEVINE, S. AND EHRLICH, S. 1991. The Freestyle Systems: A design perspective. In *Human Machine Interactive Systems*, A Klinger, Ed. Plenum Press, New York.

LITMAN, D. AND HIRSCHBERG, J. 1987. Now let's talk about Now: Identifying cue phrases intonationally. In *Proceedings of the 25th Meeting of the Association for Computational Linguistics.* ACL Press, Morristown, N.J., 163–170.

LUFF, P., HEATH, C., AND GREATBATCH, D. 1992. Tasks-in-interaction. Paper and screen based documentation in collaborative activity. In *Proceedings of Conference on Computer Supported Cooperative Work.* ACM Press, New York, 163–170.

MACKAY, W. AND PAGNINI, D. 1994. Video Mosaic: Laying out time in a physical space. In *Proceedings of ACM Multimedia.* ACM Press, New York, 165–172.

MALONE, T. 1983. How do people organize their desktops? Implications for the design of office information systems. *ACM Trans. Office Inf. Syst. 1,* 1, 99–112.

MANDER, R., SALOMON, G., AND WONG, Y. 1992. A "pile" metaphor for supporting casual organization of information. In *Proceedings of CHI'92 Human Factors in Computing Systems.* ACM Press, New York, 627–634.

MANTEI, M., BAECKER, R., SELLEN, A., BUXTON, W., MILLIGAN, T., AND WELLMAN, B. 1991. Experiences in the use of a media space. In *Proceedings of CHI'91 Human Factors in Computing Systems.* ACM Press, New York, 203–209.

MINNEMAN, S. AND BLY, S. 1991. Managing a trois: A study of a multi-user drawing tool in distributed design work. In *Proceedings of CHI'91 Human Factors in Computing Systems.* ACM Press, New York, 217–224.

NARDI, B., KUCHINSKY, A., WHITTAKER, S. LEICHNER, R., AND SCHWARZ, H. 1996. Video-as-data: Technical and social aspects of a collaborative multimedia application. *Comput. Supp. Coop. Work 4,* 73–100.

NARDI, B., SCHWARZ, H., KUCHINSKY, A., LEICHNER, R., WHITTAKER, S., AND SCLABASSI, R. 1993. Turning away from talking heads: An analysis of "video-as-data." In *Proceedings of CHI'93 Human Factors in Computing Systems.* ACM Press, New York, 327–334. Reprinted in *Information Superhighways: Users and Futures,* S. Emmott, Ed. Academic Press, New York, 1995.

NEWMAN, W. AND WELLNER, P. 1992. A desk supporting computer-based interaction with paper documents. In *Proceedings of CHI'92 Human Factors in Computing Systems.* ACM Press, New York, 587–592.

NUNAMAKER, J., DENNIS, A., VALACICH, J., VOGEL, D., AND GEORGE, J. 1993. Electronic meeting systems to support group work. In *Groupware and Computer Supported Co-operative Work,* R. Baecker, Ed. Morgan Kaufman, San Mateo, Calif.

O'CONAILL, B. AND FROHLICH, D. 1995. Timespace in the workplace: Dealing with interruptions. In *Companion Proceedings of CHI'95 Human Factors in Computing Systems.* ACM Press, New York, 262–263.

O'CONAILL, B., GEELHOED, E., AND TOFT, P. 1994. Deskslate: A shared workspace for telephone partners. In *Companion Proceedings of CHI'95 Human Factors in Computing Systems.* ACM Press, New York, 303–304.

O'CONAILL, B., WHITTAKER, S., AND WILBUR, S. 1993. Conversations over videoconferences: An evaluation of the spoken aspects of video mediated interaction. *Human Comput. Inter. 8,* 389–428.

OLSON, G. M., OLSON, J. R., CARTER, M., AND STORROSON, M. 1992. Small group design meetings: An analysis of collaboration. *Human Comput. Inter. 7,* 347–374.

PANKO, R. 1992. Managerial communication patterns. *J. Organ. Comput. 2,* 95–122.

REICHMAN, R. 1985. *Getting Computers to Talk Like You and Me.* MIT Press, Cambridge, Mass.

RICE, R. AND SHOOK, D. 1990. Voice messaging, co-ordination and communication. In *Intellectual Teamwork,* J. Galegher, R. Kraut, and C. Egido, Eds. Lawrence Erlbaum Press, Hillsdale, N.J.

ROUNCEFIELD, M., HUGHES, J., RODDEN, T., AND VILLER, S. 1994. Working with “constant interruption”: CSCW and the small office. In *Proceedings of Conference on Computer Supported Cooperative Work.* ACM Press, New York, 275–285.

SACKS, H., SCHEGLOFF, E., AND JEFFERSON, G. 1974. A simplest systematics for the organization of turn-taking in conversation. *Language 50,* 696–735.

STEFIK, M., FOSTER, G., BOBROW, D., KAHN, K., LANNING, S., AND SUCHMAN, L. 1987. Beyond the chalkboard: Computer support for collaboration and problem solving in meetings. *Commun. ACM 30,* 1, 32–47.

SUCHMAN, L. AND WYNN, E. 1984. Procedures and problems in the office. *Office: Tech. People 2,* 133–154.

TANG, J. 1991. Findings from observational studies of collaborative work. *Int. J. Man-Mach. Stud. 34,* 143–160.

TANG, J. AND ISAACS, E. 1993. Why do users like video: Studies of multimedia-supported collaboration. *Comput. Supp. Coop. Work 1,* 163–196.

TANG, J. AND RUA, M. 1994. Montage: Providing teleproximity for distributed groups. In *Proceedings of CHI'94 Human Factors in Computing Systems.* ACM Press, New York, 37–43.

TANG, J., ISAACS, E., AND RUA, M. 1994. Supporting distributed groups with a montage of lightweight interactions. In *Proceedings of Conference on Computer Supported Cooperative Work.* ACM Press, New York, 23–34.

WALKER, M. 1992. Redundancy in collaborative dialogue. In *Proceedings of the International Conference on Computational Linguistics.* ACL Press, Morristown, N.J., 345–351.

WALKER, M. 1993. Information redundancy in dialogue. Ph.D. thesis. Univ. of Pennsylvania.

WALKER, M. 1996. Limited attention and discourse structure. *Comput. Ling. 22,* 2, 255–264.

WALKER, M. AND WHITTAKER, S. 1990. Mixed initiative in dialogue. In *Proceedings of 28th Annual Meeting of the Conference on Computational Linguistics.* ACL Press, Morristown, N.J., 70–78.

WHITTAKER, S. 1995a. Video as a technology for interpersonal communication: A new perspective. In *IS and T/SPIE Symposium on Electronic Imaging Science and Technology: Multimedia Computing and Networks.* Vol. 2417. SPIE Press, San Jose, Calif., 294–304.

WHITTAKER, S. 1995b. Rethinking video as a technology for interpersonal communication. *Int. J. Human Comput. Stud. 42,* 501–529.

WHITTAKER, S. 1996. Talking to strangers: An empirical evaluation of factors underlying electronic collaboration. In *Proceedings of Conference on Computer Supported Cooperative Work.* ACM Press, New York, 409–418.

WHITTAKER, S. AND O'CONAILL, B. 1993. Evaluating videoconferencing. In *Companion Proceedings of CHI'93 Human Factors in Computing Systems.* ACM Press, New York.

WHITTAKER, S. AND STENTON, P. 1988. Cues and control in expert client dialogues. In *Proceedings of the Conference for the Association for Computational Linguistics.* MIT Press, Cambridge, Mass., 123–130.

WHITTAKER, S. AND SCHWARZ, H. 1995. Back to the future: Pen and paper technology support complex group interaction. In *Proceedings of CHI'95 Human Factors in Computing Systems.* ACM Press, New York, 495–502.

WHITTAKER, S. AND SIDNER, C. 1996. Email overload: Exploring personal information management of email. In *Proceedings of CHI'96 Human Factors in Computing Systems.* ACM Press, New York, 276–283.

WHITTAKER, S., BRENNAN, S., AND CLARK, H. H. 1991. Co-ordinating activity: An analysis of computer supported co-operative work. In *Proceedings of CHI'91 Human Factors in Computing Systems.* ACM Press, New York, 361–367.

WHITTAKER, S., FROHLICH, D., AND DALY-JONES, O. 1994. Informal workplace communication: What is it like and how might we support it? In *Proceedings of CHI'94 Human Factors in Computing Systems*. ACM Press, New York, 130–137.

WHITTAKER, S., GEELHOED, E., AND ROBINSON, E. 1993. Shared workspaces: How do they work and when are they useful? *Int. J. Man-Mach. Stud. 39,* 813–842.

WINOGRAD, T. AND FLORES, F. 1987. *Understanding Computers and Cognition.* Ablex Press, Norwood, N.J.

WOLF, C., RHYNE, J., AND ELLOZY, H. 1989. The paper-like interface. In *Designing and Using Computer Interfaces and Knowledge Based Systems,* G. Salvedy and M. Smith, Eds. Elsevier Press, Amsterdam.

Received September 1995; accepted November 1996