Achieving Speed in Globally Distributed Project Work

David Atkins[1], David Boyer,[1] Mark Handel[2], James Herbsleb[1], Audris Mockus[1], Graham Wills[1]

[1]Bell Labs
Lucent Technologies
263 Shuman Boulevard
Naperville, IL 60566 USA
{datkins, jherbsleb, audris, gwills}@lucent.com

[2]School of Information
University of Michigan
3218 Computing Center Building
Ann Arbor, MI USA
handel@si.umich.edu

ABSTRACT

We used a survey and an analysis of change management data to measure the extent of delay in a multi-site software development organization. Results indicated that cross-site work takes much longer and requires more people than comparable same-site work. We describe several tools we have deployed, which are designed specifically to address the issues of speed we identified. We report our strategy for introducing these tools, the lessons we learned and how we revised our strategy based on this experience, and present data on usage. Next, we describe enhanced versions of these tools that are being developed as products. We conclude with an examination of research findings that were helpful, and research issues we think should get more attention.

CSCW research has revealed many subtleties of collaborative work, including such things as the critical importance of context, common ground, incidental properties of shared artifacts, and informal communication. Distance work clearly impacts these crucial characteristics. Distance reduces the sheer amount of communication, and creates numerous problems in establishing a shared understanding.

This research has disturbing implications, since distributed projects, especially software development, have been thrust on product development organizations for a variety of reasons. Many countries, for example, require that development resources be located in that country in order to have any realistic opportunity to sell products there. There are also pressures to tap a global talent pool in the face of increased competition for technical talent. In addition, distributed development holds out the promise for lowered cost, and the hope (however unlikely) of faster cycle times with "round the clock" development strategies.

Our own work has focused specifically on speed, how it is affected by distance, and on ways to achieve greater speed in cross-site work. This research has been pursued by an interdisciplinary team, employing a strategy of empirical studies to understand problems and opportunities, as well as to investigate the effects of tool introduction, while simultaneously developing and deploying tools that address these issues.

Empirical research

In sum, our empirical studies have found the following:

Qualitative research. We conducted over 200 semi-structured interviews at 12 Lucent sites on three continents. We found a number of issues consistently reported, including

- issue resolution paralysis
- don't know what's "really" going on
- knowing who to contact about what
- difficulty of initiating contact
- ability to communicate effectively
- lack of trust, or willingness to communicate openly

The practical consequences of these issues include

- constantly surprised
- conflicting assumptions
- everything slowed down

Survey. In two administrations of a survey across four Lucent sites in Europe and Asia, we compared same-site and cross-site communication networks, demographics, patterns of communication, working relationships, communication and coordination, information exchange, and language. Among the findings about same-site versus cross-site differences

- cross-site communication networks are much smaller, and much less frequently exercised
- much longer waits for needed information, discussion, decision if what's needed is at another site
- more cross-site difficulty and lost time trying to find the right person to contact
- greater cross-site tendency to miss important information
- lack of clarity across sites about plans (but not tasks)
- less help by cross-site workers with heavy workloads (but equal help *offered* cross-site and same-site)

Modification Requests. Modification requests (MRs) are the basic unit of work in most software development organizations. An analysis of cross-site versus same-site MRs shows the following:

- MRs that cross sites take much longer than those where all the work is at a single site
- the cross-site versus same-site difference in interval is not related to the size or complexity of the MR

- the cross-site versus same-site difference in interval appears to be mediated by the number of people who work on the MR (cross-site MRs require more people than comparable same-site MRs, and the number of people is the most significant determinant of interval)
- interdependence among sites (as measured by proportion of MRs that are cross-site) is steadily increasing

Our efforts to address these issues have included a number of recommendations about communication practices and commercial tools. But here we report on prototypes of new collaboration technology deployed internally, as well as enhancements of these tools that are being incorporated into products.

Initial tools

After fairly extensive interviewing, we thought we understood the users' needs well enough to design tools aimed directly at the most severe and pressing needs. After briefly discussing the tools, we describe our initial deployment experience. The internally-deployed tools include Rear View Mirror, CalendarBot, and Experience Browser.

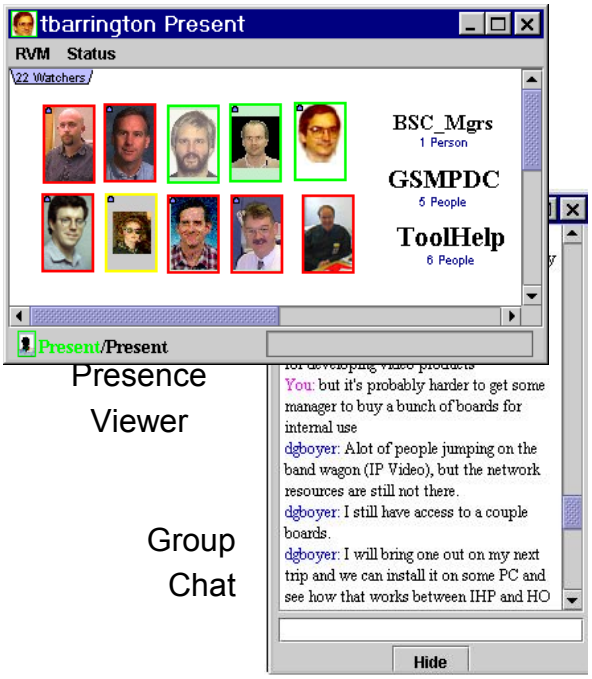


Figure 1. Rear View Mirror

Rear View Mirror (RVM) is a presence awareness and instant messaging tool. The presence viewer allows users to see who else is logged on, and permits them to initiate a chat with anyone who is present. An additional feature is the "group chat" room. For all members of the group, the group chat opens as soon as they log on. All members of the group can see all entries and add their own. The chat persists for a period of time determined by the group administrator, so that workers offset by time zones can catch up with what their earlier-rising colleagues talked about, or what their later-quitting colleagues did after the end of the local work day.

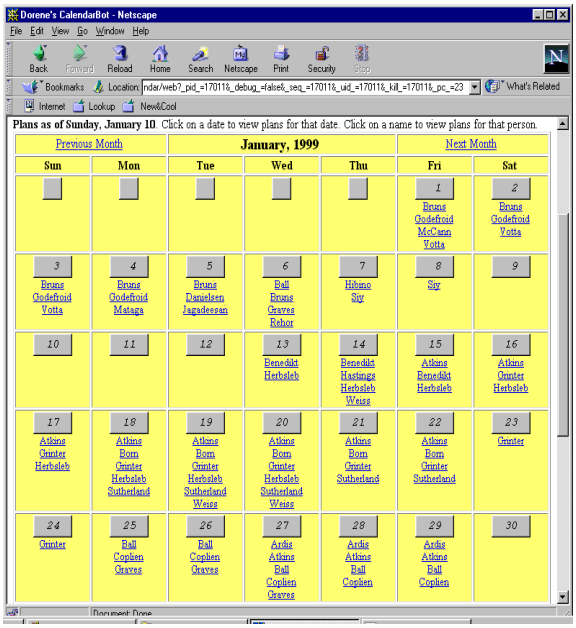


Figure 2. CalendarBot

CalendarBot is a web-based calendar tool that is primarily used for course-grained (all day or half day) information about such things as travel, vacation, and training. It has a variety of views, including calendars and lists. Users can define groups, and views can be filtered based on group membership. Entry of data is very straightforward. Anyone can administer anyone's data.

Experience Browser is a tool designed to address the need to find an expert in some particular part of the software. Since most telecommunications systems have large and highly complex software with substantial dependencies among the various parts, it is often the case that someone needs to find an expert in some part of the software he/she is not familiar with. Experience Browser allows one to select any part of the software (any subsystem, or module, or file) and with a single click, produce a list of all developers who have contributed code, in order of how much they contributed. It also allows the user to click on any individual or organization and see where this person

has contributed code. It also displays contact information for each person.

Tools meet the real world.
An initial version of these tools were used extensively by the research team, and when we were confident that they were sufficiently reliable and had a sufficiently useful set of features, we introduced them tool into a development organization.

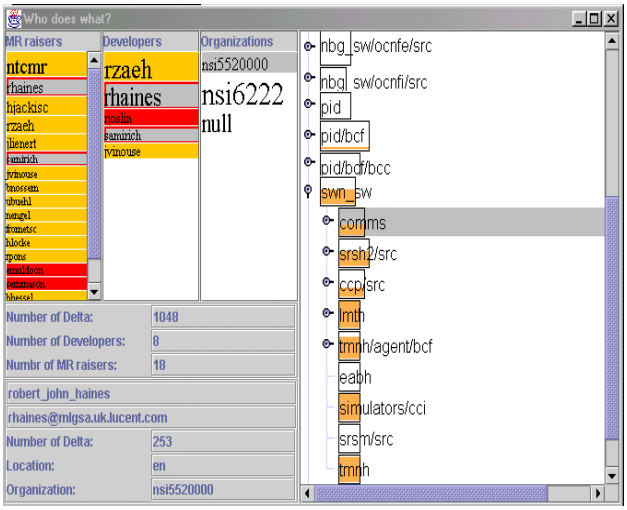


Figure 3. Experience Browser

Phase 1. Our plan for introducing tools was based on a strategy for achieving critical mass. We decided to introduce tools to a single communication "pipe," (i.e., those individuals at two different sites who frequently communicate with each other) and to do so by sending people to both sides of the pipe at once. Based on a survey item that asked who people communicated with at other sites, we identified what seemed to be the "central" dyads of people, i.e., all those pairs of people who are most frequently named by people at the other site as people they communicate with, making sure that those pairs that communicate most frequently are included.

After identifying our targets, the director, i.e., our sponsor in the development organization, sent out an e-mail announcement about the upcoming tools installation and training effort. We then scheduled 1-hour slots with as many of the central people as we were able, and sent 2 people to each site. Installation and training was done over a 4-day period at both sites. There was then a period of 1 additional week of follow up at both sites, with someone calling in person on each of our installees.

We prepared for installation in a fairly elaborate way. We identified the operating systems being used at each site, the version of java jdk installed, prepared quick reference cards for each tool, had a tool web site with pointers to the cards and e-mail addresses for help. We had installation CDs of all needed software (including browser plugins, etc.) so installation would be fast.

The results were reasonably encouraging. CalendarBot quickly became a runaway success, with many thousands of hits each month. The Experience Browser had a relatively low but consistent use pattern, which is probably what one would expect for an expertise location tool, since it is not something one needs on a daily basis. Rear View Mirror, perhaps the tool most sensitive to critical mass issues, had a fairly high initial use, which fell off quickly to a small core of users. RVM also suffered from reliability and usability problems.

About one month later, with RVM reliability much improved, this process was repeated with a second pipe. The second pipe involved one of the same sites as the first, but an almost entirely different set of people, since people tended to interact primarily with one distant site or the other, and not both.

The most obvious lessons from this experience were:

1. *Casual use is not testing.* The research team had used the tools for about two months, and all the major problems had bee fixed. Unfortunately, the functionality exercised most by the research team was not necessarily the functionality that would be used most by the development group. For example, the research team basically created and used one group in RVM, with a few others created just to make sure that worked. The developers, on the other hand, created enormous numbers of groups, shifted people in and out of them, removed users from groups while those users were logged into the group and in the middle of chatting, etc. These "strange" behaviors unearthed many bugs, much to the users' distress. The obvious solution was more testing of subsequent versions.

2. *Awareness or surveillance.* Particularly in Germany, many workers were concerned that RVM could be used for surveillance. In fact, we were told it would present huge problems in the workers' council, and that it might even be illegal. We heard these concerns well before deployment, and implemented a security policy that would permit only those explicitly allowed by a user to see that user's presence information. This, unfortunately, had the effect of making installation and setup nightmarish, since after installation, the user could not "see" anyone else, but would have to cycle through all his/her associates and grant each one permission. Each of these associates would then have to grant *this* user permission to see *them*. With people under great time pressure, this whole procedure was much too cumbersome. The solution we have adopted is to base permissions on group membership. If you are in a group, then you can "see" all of the members and they can "see" you. Others can be permitted on the (more cumbersome) individual basis. (We still don't have many German users.)

3. *Generating critical mass.* Critical mass issues arose in several forms. CalendarBot presented relatively straightforward problems, in that the calendar has no value until a number of users entered data. But no one felt like bothering to enter data, since the tool was never used. This issue was successfully addressed by having secretaries and administrative assistants enter data. They were able to create enough interest that tool use took off, and several CalendarBots are getting many thousands of hits each month.

The stickier critical mass problem was presented by RVM. As mentioned above, we tried to anticipate this issue by sending people to two sites simultaneously to install and train people on the software, and by using survey data to identify the people that communicated most frequently across specific pipes.

The strategy was not terribly successful. As we realized later, critical mass for certain tools exerts its effect *by group*, not on the organization as a whole. One needs near-saturation among members of a group who will use the tool with each other. Having socially "central" but thinly spread dyads of users was not effective. Focusing on distributed groups has been a more successful strategy. Installing and training all members of a group *simultaneously* is very important – they can begin to interact through the tool right away.

4. *Installation & administration issues.* Despite our preparations, there were some installation problems. For example, the Experience Browser required a java 1.2 plugin. Yet one of the sites primarily used an older version of unix for which there was no such plugin. Fortunately, the team was able to improvise and provide a workaround.

Administration proved to be nightmarish. On the theory that having a server installed locally would improve performance, we installed the RVM server in Swindon, and hired a contractor to reside there, do testing, user studies, marshal support and interest, and take over some coding duties. The only small problem, as it turned out, was that the CIO, which finally gave its approval to install a server on their LAN, kept the server locked in a closet and would not give our contractor a key. Furthermore, there was no one on site with a key. Entry had to be arranged days in advance. Worse, we were not permitted to install any remote administration tools. So if we needed to add a new user, or start up a crashed or flaky server, we were simply out of luck until they got around to finding a key and letting us in. We have since moved the server to Naperville (which was not easy, even with our small installed base of users).

We track tool use with on-line logs and visualization displays. Typical current use for these tools is ~3,000-3,500 hits per month for CalendarBot (representing about 350 potential users); about 50 hits per month on Experience Browser (about 350 potential users), and about 30 RVM logins per day (about 190 potential users).

From prototype to product

Based on our research identifying factors that seem to hinder multi-site projects, and the technology developed for internal use to address those problems, we speculated that most large corporations had similar issues. If true, then these technologies could form the basis for future products.

Moving toward products introduces many additional constraints, e.g., where does Lucent have potential advantages and how can we exploit these? In our case, Lucent has lots of experience with voice networks, switching, and a broad product range in both wireless and wireline. It is to our advantage to have features in the network, rather than in the endpoints (Lucent sells network equipment, not end user equipment.) Finally, we, like most similar companies, are looking for ways to exploit converged networks, i.e., networks where voice, data, video, etc., are all on the same (probably IP) network.

The research group is collaborating with a development group that wants new and interesting features for a Lucent product called Softswitch Programmable Feature Server (SPFS). Integration with SPFS adds new possibilities to all the tools. Specifically, SPFS gives us the capability to flexibly add in audio features. We began working with a systems engineer to extend the tools' functionality, to design tools that implement ideas we had but did not know how to build, and to construct scenarios to show the utility of the ideas. A team of researchers, developers, architect, systems engineer, project manager, and product manager was assembled.

The initial prototypes ranged from existing collaboratory tools to ideas implemented only in PowerPoint. The initial activity followed two parallel paths, marketing and development. The initial presentation was heavily decorated with clip art, charts about markets, and other marketing paraphernalia. Customers were identified, and presentations to internal customer teams and eventually external customers began.

The other activity was the development itself. The team began identifying the requirements for the new tools and also began systematically reviewing other existing technology from research and other sources to see if could be incorporated in order to speed up development.

What follows are screen shots and descriptions of the technology that will soon be undergoing market trials, as well as trials "under the microscope" inside Lucent. As of this writing, the tools are undergoing usability testing by means of cognitive walkthroughs and heuristic evaluations.

TeamPortal. Team Portal allows a worker to select, from a company directory or a local address book, a set of people that he or she needs to stay closely in touch with (see Figure 4). For this set of people, TeamPortal displays the right set of world clocks, as well as world calendars with holidays for the relevant countries. With a single click one can also access the permitted entries of each person's individual calendar.

The bottom panel is where the application begins to take advantage of convergence, i.e., having all data (audio, web, etc.) flow through one network. For each person, a set of icons to the right of the person's name indicates recent activity (or inactivity) of the person's devices. One might see, for example, that Audris used his office phone about 15 minutes ago, is currently using his desktop computer, and has his cell phone switched off. If you need to reach Audris, a single click on his office phone icon will, depending on your and his individual preferences, connect you by softphones or regular POTS (Plain Old Telephone System) phones in your offices.

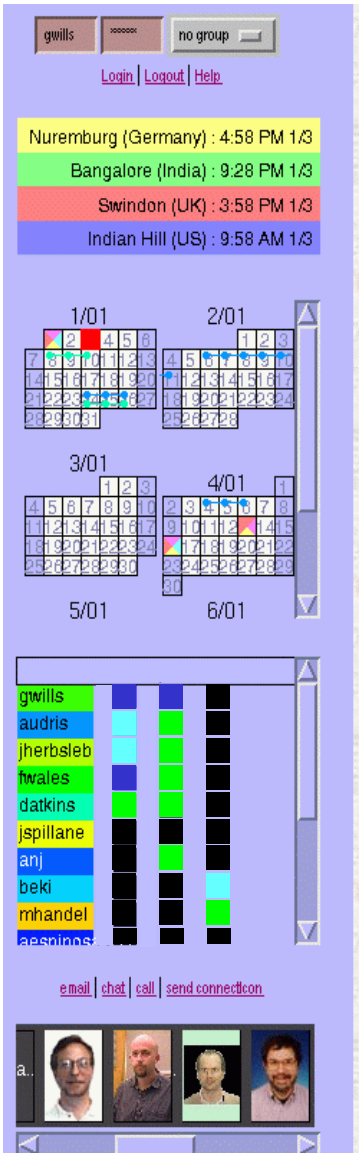


Figure 4. Team Portal.

Mhandel, on the other hand, has shown no desktop computer activity today, and has not taken or made a call from his office phone. His cell phone is on, however, so you can place a conference call to Audris and Mhandel from the desktop by selecting both of the appropriate icons. If one or both of them is unavailable, you can set an alarm, and TeamPortal will notify you as soon as both of them are available in the way you need, e.g., anything that will support an audio connection.

TeamPortal also supports these capabilities beyond your current team, or set of close contacts. Suppose you need to contact someone on the marketing team for a particular product. You navigate to the team's web page, and with a single click, TeamPortal displays the correct world clocks, calendar, and presence information for the team. If the team has designated a contact person, or a policy for specifying a contact person, only that person's contact and presence information is displayed.

ConnectIcon. The purpose of a ConnectIcon is to make it possible to talk with someone about a specific topic or issue as quickly as possible, i.e., as soon as all the relevant people are present, available, and prepared. The ConnectIcon allows communication that closely approximates this ideal – it is intended to be an antidote to phone tag.

The sender of a ConnectIcon configures it with a sentence or two describing the topic of the intended interaction, the identity of the ConnectIcon receivers, and URLs for any

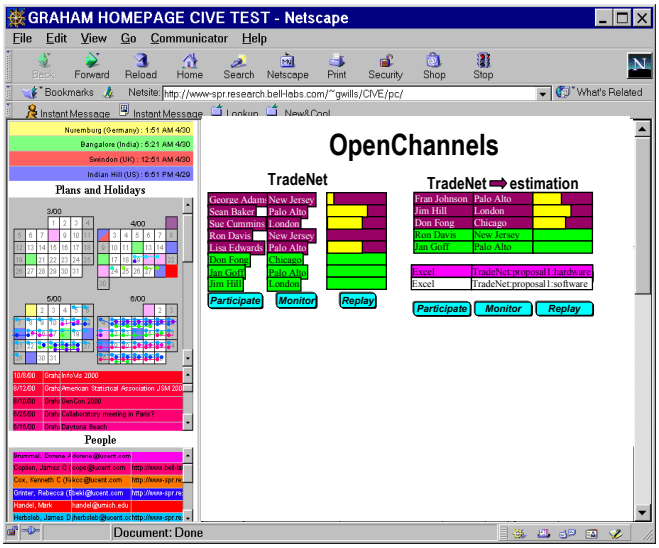


Figure 5. Open Channel.

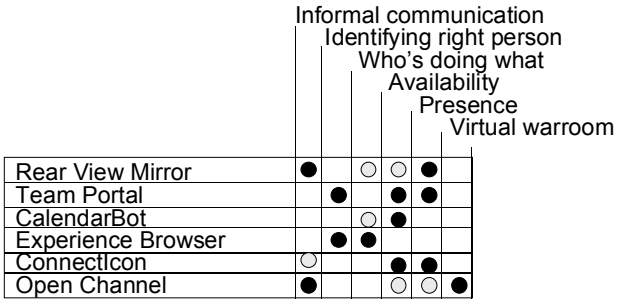
relevant materials. If the receiver also has TeamPortal, the ConnectIcon is displayed in the TeamPortal interface. A mouse-over displays the text message, and a right click generates a popup menu that allows the receiver to access any of the URLs specified by the sender, access the sender's calendar, and initiate a chat, e-mail, or audio connection. In addition, the ConnectIcon also displays presence data, allowing the receiver to see if it is a good time to initiate contact to take care of the task. As soon as the receiver is prepared, and the sender is available, contact can be initiated.

For those without TeamPortal, the ConnectIcon is delivered as a URL, generally as an e-mail attachment. Clicking it brings up a small, iconified picture of the sender, and provides the same functionality as above.

OpenChannel. For co-located colleagues working closely together, particularly in stressful or crisis situations, their ability to stay closely in touch just by talking to each other is a huge advantage. In fact, some organizations have experimented with locating everyone in a single room, a "war room," as a way of speeding up difficult tasks requiring coordinated activity. In our interviews, we heard about a number of occasions where conference bridges were kept up almost continuously for as long as 5 weeks in order to allow a global team to work together on a critical problem. By all reports, despite the many shortcomings of conference calling, it was seen as very beneficial. These observations, along with the published research on voice loops (e.g., [7] and co-located software "war rooms," [14] led us to the Open Channel idea.

Creating long-standing audio connections can create a war-room like functionality. In some respects, of course, it is inferior since it lacks large display areas and the easy establishment of common ground that co-location provides. It nevertheless has the advantage that whoever is needed, from anywhere in the world, can participate in ongoing work. Anyone can ask a question, or describe a new finding just by talking. For many problems, resolution can be accelerated by passing problem solving off between widely separated time zone, with OpenChannel providing the means for the new people to get up to speed during whatever overlap in working hours is available.

The basic long-standing audio connection is enhanced by placing all participants in an application-sharing session so they can jointly view and edit documents, or whatever other applications that might form the basis of collaborative activity. In addition, Team Portal gives participants the opportunity to turn off the sound, if it becomes bothersome, and monitor the audio channel visually (see figure 5). The visualization shows who is participating in the OpenChannel, who has spoken recently, and what applications are open in the application sharing session. One can listen to and/or visually monitor several channels at once.



Finally, OpenChannel provides recording and replay capability. The entire session, including application-sharing activity with pointing, scrolling, and editing, can be captured, and replayed at original or accelerated speed.

In addition, a participant who is monitoring visually may, on the basis of who is speaking and/or what documents they are sharing, decide to replay the last few minutes to see if the topic is of interest, and he/she can decide whether to join more actively.

Together, these three services provide ways of finding people and connecting with them as soon is it is possible fruitfully to do so. They provide a channel for extended, background communication as well as foreground communication. A mapping of tools and functionality is provided in the figure above.

Research and The Real World

In this section, we list a number of real-world issues raised by this project, which will provide grist for theme-oriented discussion.

Research we used.

There has been much research on specific kinds of tools that we found very useful, e.g., calendars [13], audio channels [7], instant messaging [11], chat [2], expertise location [10], MUDs [3] and awareness [4, 5], to name just a few.

There was also research on co-located work that we found very relevant, both for understanding the potential issues in cross-site work by recognizing the various subtleties of co-located work (too many to cite), and possibilities for recreating at least some aspects of co-locates work with new technology (e.g., radical co-location [14], and informal communication [9]).

We also profited from work on distance and collaboration, e.g., [1, 8], and the very limited research on using sets of tools together to actually accomplish real-world tasks, e.g., [12]. The work on why groupware fails, succeeds [6] was quite helpful, but the research we've seen in this vein doesn't go nearly far enough. Although many of the most important general considerations are now well known, we still seem to know very little about the specifics of when and how which application will fail or succeed, and how to tip the balancd

Less useful than we would have thought.

Research on synchronous work generally does not directly address the most important pragmatic issues, from our point of view. Meetings are awkward, but generally bearable and "good enough." In the overall context of a project, meetings are unpleasant, but they don't often seem to be the source of delay.

Research directions we would like to see pursued more vigorously:

Reliability / requirements discovery dilemma. There is consensus that the "real" requirements for collaboration

tools are nearly impossible to determine correctly in advance, and intuitions are notoriously inaccurate. Yet users have very little tolerance for unreliable, unavailable tools. How do you generate rapidly modifiable, robust, tools for actual use in trials?

Possibilities:

- find extremely tolerant users
- plan to lose some user groups
- huge investment in prototypes
- frameworks for rapid prototyping

Introducing collaboration tools into an organization. The logistics of who gets trained, how, and on what; how the tool is positioned to them; the role of management, etc. How can one maximize the chances for adoption? The issues include

- Reaching critical mass.
- Integrating with other tools, environment
- Incentives
- Staging introduction
- Sustaining use through organizational change
- Cultural differences.

Research that takes a broader view of longer-term tasks. One can find much good research on meetings, with or without video, etc., research on use of calendars, use of video, instant messaging, and so forth. How are these all used together to get work done? How does a project team, for example, modulate between synchronous and asynchronous activities? How does one use asynchronous technologies, for example, to get a group of people together for an effective synchronous session as quickly as possible? How can tools break down the meeting / not meeting dichotomy so that interaction can occur flexibly, with natural interleaving of various group and individual activities?

What is the business value of collaboration tools? One must generally make a pretty strong case for the out-of-pocket cost, but also for the disruption that inevitably accompanies adopting a tool. It is also important to have some idea of business value to set priorites. Should we devote limited attention and resources to an online calendar tool? Instant messaging? Better speaker phones? Or something completely different, like hiring another person to help with the work rather than investing in collaboration tools?

Non-researchers as users. Research is very different from, e.g., project work, in terms of degree of interdependence (researchers generally much less interdependence), time pressure (generally much less), tolerance for suboptimal software (much more), and willingness to try something just out of curiosity (much more). These factors make it very difficult to generalize from researchers to non-researchers.

REFERENCES

1. Allen, T. J., Managing the Flow of Technology. 1977, Cambridge, MA: MIT Press.

2. Bradner, E., W. A. Kellogg, and T. Erickson. The adoption and use of "Babble": A field studyof chat in the workplace. in *ECSCW* 1999. Copenhagen, Denmark: Kluwer 139-158.

3. Churchill, E. F. and S. Bly. It's all in the words: Supporting work activities with lightweight tools. in *GROUP '99*. 1999. Phoenix, AZ 40-49.

4. Dourish, P. and S. Bly. Portholes: Supporting Awareness in a Distributed Work Group. in *CHI'92,* 1992. Monterey, CA.: ACM Press 541-547.

5. Fish, R. W. Design of a Multi-Media Vehicle for Social Browsing. in *CSCW '88,*. 1988 25-38.

6. Grudin, J. Why CSCW applications fail: Problems in the design and evaluation of organizational interfaces. in *CSCW '88*. 1988. Portland, Oregon: ACM Press 85-93.

7. Hindus, D., *et al.* Thunderwire: A field study of an audio-only media space. in *CSCW* 1996. Boston, MA: ACM Press 238-247.

8. Kraut, R. E., C. Egido, and J. Galegher, Patterns of Contact and Communication in Scientific Research Collaboration, in *Intellectual Teamwork: Social and Technological Foundations of Cooperative Work*, J. Galegher, R.E. Kraut, and C. Egido, Editors. 1990, Lawrence Erlbaum Associates: Hillsdale, NJ. 149-171.

9. Kraut, R. E., *et al.*, Informal communication in organizations: Form, function, and technology, in *People's Reactions to Technology*, S. Oskamp and S. Spacapan, Editors. 1990, Sage: Newbury Park. 145-199.

10. McDonald, D. W. and M. S. Ackerman. Expertise Recommender: A Flexible Recommendation System and Architecture. in *CSCW* 2000. Philadelphia, PA: ACM Press 231-240.

11. Nardi, B. A., S. Whittaker, and E. Bradner. Interaction and Outeraction: Instant Messaging in Action. in *CSCW* 2000. Philadelphia, PA: ACM Press 79-88.

12. Olson, J. S. and S. Teasley. Groupware in the Wild: Lessons Learned from a Year of Virtual Collocation. in *CSCW* 1996. Cambridge, MA: New York, N.Y.: ACM Press 419-427.

13. Palen, L. Social, individual, and technological issues for groupware calendar systems. in *CHI '99*. 1999. Pittsburgh, PA

14. Teasley, S., *et al.* How does radical collocation help a team succeed? in *CSCS 2000*. 2000. Philadelphia, PA: ACM Press 339-346.