Meeting Central:
Making Distributed Meetings More Effective

Nicole Yankelovich, William Walker, Patricia Roberts, Mike Wessler, Jonathan Kaplan, Joe Provino

Sun Microsystems
One Network Drive
Burlington, MA 01803
+1-781-442-0441

[*firstname.lastname*]@sun.com

ABSTRACT
The Meeting Central prototype is a suite of collaboration tools designed to support distributed meetings. The tools' minimalist design provides only those features that have the most impact on distributed meeting effectiveness. The collaboration suite is built on top of a distributed, extensible, and scalable framework.

Categories and Subject Descriptors
H.5.3 [**Group and Organization Interfaces**]: Collaborative Computing – *synchronous interaction, CSCW, organizational design.*

General Terms
Design, Experimentation, Human Factors.

Keywords
Distributed meetings, audio conferencing, group communications, collaboration framework, distributed system, VoIP.

1. INTRODUCTION
Meeting Central is focused on improving the effectiveness of distributed meetings. We define *distributed meetings* broadly to mean two or more people in different locations using any form of communication. It could be, for example, a quick phone call, an instant message, an ad hoc meeting with a few people, a regularly scheduled group meeting, or a large formal meeting. This broad definition means that we are concerned not only with users' activities during the meeting itself, but also the mechanisms for looking up people and groups, and the means to establish communication with them.

The Meeting Central project involves a number of aspects. As part of this project we have conducted internal and customer user research, we have designed a suite of distributed meeting tools, and we have created an open architecture for collaboration-aware applications that includes the concept of a *SPOC* or *single point of contact.* This paper describes highlights from the user research, gives a flavor for the Meeting Central user interface, and includes a high-level description of the architecture.

CSCW '04, November 6–10, 2004, Chicago, IL, USA.

2. RELATED WORK
The body of literature related to topics touched by Meeting Central is extensive. We highlight below those projects that are most closely related to Meeting Central or that have had a particular impact on our thinking.

2.1 Distributed Meeting Tools
Many other research groups have tackled the problem of distributed meetings. A few examples of important work in this area include projects such as Forum [13], Thunderwire [1], Rapport [2], Project Nick [8], and TeamSpace [11]. There are also a growing number of commercial products in this space. Some of the more widely used tools include Microsoft Live Meeting, WebEx Meeting Center, Macromedia Breeze, Lotus Sametime, and Centra 7. These tools, along with many others include a large collection of features including follow-me browsing, annotations, application sharing, synchronized presentations, floor control, shared whiteboards, voting, polling, Q & A, and so forth. An example of every feature in Meeting Central can undoubtedly be found in some previous project. What differentiates Meeting Central is not the uniqueness of the individual features, but rather the careful selection of features. As is explained in the User Research section below, what we discovered is that complexity is a major problem. People want to focus on the content of their meeting, not on the meeting tool. Our particular combination of features was chosen, based on an analysis of user data, to maximize meeting effectiveness while minimizing complexity.

2.2 Augmenting Audio Conferencing
One example of how we chose to minimize complexity is illustrated by comparing Meeting Central to the SmartPhone system from University of New South Wales [15]. The goals of the two projects are similar in terms of adding what Tim Moors calls "a complementary control channel" to an audio conference. The Meeting Central Facilitator and the SmartPhone both provide some visual communication such as hand raising and expression buttons. The systems differ dramatically, however, in the amount of structure they impose on a meeting. With the SmartPhone, users are in one of four states: Listening, Queuing, Interacting and Leading. In addition, there is a Convenor role which includes the ability to selectively mute participants. In Meeting Central, no roles are imposed. Interestingly, when Moors describes some user experience with the SmartPhone, he discovers that "Users tended not to adhere to the displayed speaker queue, but rather used it to create awareness of who wanted to speak, with social protocols determining who actually spoke." Meeting Central starts with the premise that all actual control is socially mediated. As Erickson and Kellogg point out, conversation is a "fundamentally social process" [10]. The software "control channel" is provided to facilitate the social mediation, not supplant it. This observation

has allowed us to eliminate all the user interface complexity that comes from imposing states and roles. In Meeting Central, for example, there is no concept of a Presenter role. Any user can just start presenting. We count on the power of social convention and socially defined roles to prevent users from usurping control of a presentation when it is inappropriate to do so.

A more recent project called Enhanced Telephony demonstrates person to person calling along with the ability to share documents using Microsoft NetMeeting or start an instant message exchange between the participants [6]. This work, along with our previous work on Awarenex [21], focuses primarily on contacting others and integrating communication channels into the user's computing environment. While Meeting Central includes similar functionality, the focus is on augmenting the real-time conversation that happens after contact has been made.

Also on the topic of augmenting real-time audio communications, a study by Alex Colburn, et al looked at two forms of graphical enhancements for audio conference calls. One involved highlighting static photographs of the current speaker and the other involved 3D avatars with head movements. Interestingly, "Experiments show that both graphically enhanced interfaces improve the understandability of conversations, particularly with respect to impressions that others in the group could express themselves more easily, knowing who is talking, and when to speak. Little difference was found between the two interfaces." [7]. While we have no empirical evidence yet to confirm it, our belief is that Meeting Cental's more sparse graphical representation of the person speaking (described in section 4.3) improves the understandability of audio conversations in the ways demonstrated in the Colburn study. One possibility is that it is not graphics that trigger the benefits of increased understandability, but rather the act of binding identity to the person speaking that does the trick.

2.3 Collaboration Frameworks

A number of other research groups have created frameworks to help software developers create collaboration-aware applications. One of the most notable groups in this regard is the GroupLab at the University of Calgary, having produced collaboration toolkits such as GroupKit [12], and collaboration frameworks such as TeamRooms [19]. The BSCW System [5] provides an interesting web-based approach. Other efforts include Java-technology approaches such as Flexible JAMM [4] and Sieve [14].

Rather than adding a new collaboration toolkit to this set of approaches, the Meeting Central architecture focuses on secure, distributed and extensible *collaborations* that can be implemented in flexible ways. Because of this flexibility, individual collaborations can implement or interact with one or more of the aforementioned collaboration frameworks. For example, we envision that a Meeting Central collaboration could use Flexible JAMM to allow multiple users to simultaneously enter text in a text area. In the same light, we defer to the individual collaboration implementations to decide how to manage the replication of data and synchronization of user events.

In this respect, perhaps the closest approach to our work is that of TeamRooms. We differentiate our work by providing tight telephony integration, supporting the recording of events that take place over time (although TeamRooms could be extended to do this), managing privacy and security, and taking a more decentralized approach.

Furthermore, our architecture is notable in that it is a Java™ technology development platform that allows application developers to create platform-independent tools. These tools include rich interactive clients as opposed to dynamic HTML content.

Although the system is based on the Java platform, it is not restricted to solutions based solely on Java technology. For example, our voice conferencing bridge interacts with telephony systems using SIP, and we are currently creating an instant messaging collaboration that communicates with an instant messaging server using the XMPP protocol.

3. USER RESEARCH

What problems are people currently facing that cause distributed meetings to be ineffective? This was the primary question we wanted to understand before embarking on the design of a distributed meeting tool.

One goal of our user research was to establish several baseline measures that would allow us to determine the effectiveness of the Meeting Central collaboration suite and how much the tools impact people's ability to conduct productive meetings.

Another aim was to understand more about the nature of existing distributed meetings. For example, how happy are people with their current way of conducting meetings? How often are conference rooms involved? How frequently do people attend meetings face-to-face versus remotely? Do people judge effectiveness differently if they facilitate the meeting?

3.1 Data Sources

For this project, we conducted an internal study of our company's employees to form the basis of the software design and to provide baseline data. After designing our initial prototype, we launched an external customer study to validate the generalizability of our internal data, to collect competitive data, and to examine some issues raised during the prototype development process.

3.1.1 Internal Study

For the internal study, we designed a survey to measure current levels of distributed meeting effectiveness and to understand what distributed meetings problems were most severe. In July of 2002, we emailed the survey to 8,500 employees and received 1,782 responses for a 21% response rate.

By their nature, most survey responses are subjective. To verify our findings, we validated our results with additional data that was more objective. We started by analyzing two years worth of trouble tickets that had anything to do with distributed meetings. Here we counted the number of times different types of problems were mentioned. We then reviewed two years worth of postings to six discussion aliases that covered specific distributed meeting tools such as "`VNC users`" and general topics related to distributed work like "`work-from-home`." Again, we cataloged problems mentioned and recorded their frequency.

Our internal investigation covered one additional data source. We examined training material for two mandatory courses that all employees were expected to complete. In these, we were looking for methodologies being taught that were difficult to accomplish without face-to-face interaction. For example, both classes teach teams to conduct affinity-style brainstorming, organizing and voting on ideas with sticky notes.

3.1.2 External Customer Study

After we prototyped the software described in the following section, we conducted a second study with external customers. For this study, we first interviewed 18 customers or prospects. These people fell into one of two categories: those that participated in at least one distributed meetings per week, and those that administered tools to support distributed meetings.

Using the data collected from the interviews, we designed a customer survey to assess the generalizability of our internal results and to assess the effectiveness of people's existing meeting tools. In February of 2004, we emailed this survey to 5,216 customers and prospects and received 325 responses for a response rate of 6.2%.

3.2 Findings

3.2.1 Internal Study Findings

The internal survey yielded a rich collection of data. From the survey, we compiled a list of the top problems that people have in distributed meetings. To make it onto this list, the problems had to be both frequent and correlated with meeting effectiveness.

Top Problems with Distributed Meetings

Audio Problems
Some people could not be heard 33.9% P=.004
Poor audio quality 23.8% P=.001
Too much extraneous noise 20.2% P=.000

Behavior Problems
Speakers did not check others for understanding 11.3% P=.000
Participants did not follow effective mtg behaviors 8.6% P=.008
Inadequate advance planning 7.4% P=.000
Meeting was not well facilitated 6.0% P=.000

Technical Problems
Difficult to identify who was speaking 29.1% P=.000
Not everyone could view visual materials 18.0% P=.000
Not tell who was in mtg, who joined, who left 13.4% P=.005
Necessary documents not available during meeting 6.9% P=.001

We grouped the problems into three categories: *audio, behavior,* and *technical.* In the list above, the percentages refer to those respondents who indicated that this problem had occurred in a recent distributed meeting. The P values represent a confidence score. A value of .000 indicates the highest level of confidence that the problem contributes to people's assessment of meeting effectiveness. "Dropped from conference call" is an example of a problem that occurred as frequently as some of the others (8% of respondents reported dropped calls), but that was not strongly correlated with meeting effectiveness (P=.378).

The audio problems were the most frequent and had the largest negative impact on meeting effectiveness. Since humans are particularly sensitive to auditory input, they tend to rate a meeting as less effective if the audio quality is compromised by excessive background noise or insufficiently mic'd speakers.

While behavior problems did not occur as frequently, they were highly correlated with meeting effectiveness. We therefore spent some time unpacking what people meant by "not following effective meeting behaviors." We conducted a few focus groups and reviewed the free-form survey comments associated with this option. We discovered that people have the following sorts of behaviors in mind: attendees not identifying themselves when they speak, presenters not indicating the current slide number, facilitators not checking for audio problems, people not moving close enough to microphones, and people gathered in a conference rooms generally ignoring remote participants.

Of the technical problems in the list above, two are closely related: "Not everyone could view visual materials" and "Necessary documents not available during meeting." There were a multitude of reasons why visual materials were not available or could not be viewed. In some cases the presenter did not send a pointer to the materials to remote participants, in some cases the presenter did, but sent them so late that the remote attendees did not get them before leaving their offices. In other cases people did not have easy access to an application to view the materials.

The other two technical problems in the list can be characterized by not knowing what is going on in a meeting – not knowing who is speaking, who has arrived, or who has left.

Here are some additional highlights from the internal survey.

Data Conferencing: Only 151 respondents (8%) had used any form of data conferencing tool. Of these people, 35.5% used them for presentations. Other uses were sharing web pages (19.2%), jointly editing documents (18.7%), demonstrating applications (12.7%), and whiteboarding (9.6%).

Meeting Size: Most meetings included 6-9 participants (46.9%). Meetings of 2-5 and 11-16 participants were also common (21.5% and 19.4% respectively). The larger the meeting size, the lower the meeting effectiveness. More meeting problems were reported with meetings of 11 or more participants than those with fewer participants.

Meeting Duration: Meetings lasting less than one hour were most frequent (52%) followed closely by meetings of 1-2 hours (39%). Less frequent were meetings of 2-4 hours (6%) and meetings of greater than 4 hours (3%). Meeting effectiveness was rated lowest for meetings of 2-4 hours and more meeting problems occurred in meetings lasting longer than 1 hour than in shorter meetings.

Meeting Configuration: Just under 70% of all meetings involved conference rooms. Most meetings involved one conference room with additional participants in dispersed locations (58.8%) and some involved two or more conference rooms (10.2%). The remaining 31% of meetings took place with all participants distributed. A higher number of problems were reported in meetings held in 2 or more conference rooms than in meetings with fewer than 2 conference rooms.

Participant Location: Participants were most often located with at least some other people (56.7%). Those participating on their own (43.3%) reported significantly lower overall meeting effectiveness than those participating with others.

Participants vs. Facilitators: Those people who facilitated or ran meetings (28%) consistently reported the meetings as more effective than people who were participants (72%).

Meeting Types: The most frequent meeting purposes (in order) were: general staff meetings, planning meetings, project reviews, transfer of information (TOI) meetings, and decision-making meetings. Also common were project team meetings, brainstorming sessions, presentations, and document reviews.

3.2.2 External Customer Study Findings

After creating a preliminary Meeting Central prototype, we began the customer portion of our user research. In our 18 customer interviews, we explored what collaboration tool features were most important to people. Consistent with our internal data, top on the list was the ability to share presentations. Also consistent with our internal data is an emphasis on technology that would work well in conference rooms with people also connected over the telephone. The need to identify who is attending, leaving and speaking in a meeting was another common theme as was being able to collaborate with people external to one's enterprise just as easily as internal people.

Ease of use was perhaps the strongest theme to emerge from the interviews. One story that recurred in three interviews involved someone attending a distributed meeting using a real-time collaboration tool and liking the experience. But when asked if they used the tool in their own meetings, they all said "no, it was too complicated to figure out how to set it up."

Since we talked to tool administrators and not just end users, a number of infrastructure requirements emerged. People emphasized the importance of high-quality audio, they wanted tools that will work from anywhere, a secure connection guarantee, and scalability.

Perhaps the most interesting finding to emerge from the interviews was the importance of organizational commitment to collaboration. We interviewed people from two different large companies that had spent a great deal of money purchasing site-wide licenses to collaboration tools. The first company sent an email out to employees announcing the availability of the tools and how to access them. The number of employees that actually used the tools was extremely low. For those that did use them, it was frustrating because they always had to train the people they wanted to collaborate with. The other company had made a substantial organizational commitment to the technology. They created an internal marketing group to evangelize the tools and they included tool training as part of new employee orientation. In that company, everyone made the assumption that their colleagues knew how to use the tools, so the tools became a standard part of doing business at that company.

In the customer survey, we asked "Overall, how satisfied are you with the existing distributed meeting resources and services that your company provides? Rate from 1 to 7, where 1 = Very Dissatisfied and 7 = Very Satisfied." The mean response of 4.15 shows that there is room for improvement in terms of tool offerings and organizational support for those tools.

The survey indicates that four tool features are significantly related to overall collaboration tool effectiveness: quality, ease of use, suitability for all different size meetings, and suitability for company-wide deployment.

One question we asked customers which we did not ask our internal respondents is why they did *not* use real-time collaboration tools in their distributed meetings. Not surprisingly, since we considered one-on-one conversations as distributed meetings, 44% said the tools were not needed. For those that thought tools were needed, the following reasons for not using them were most frequent: "they require too much work or take too much time to set-up" (29%), "they are not widely used within our company" (20%), "I can't get others to use the tools" (18%), "cost concerns" (16%), and "they are difficult to use" (15%). The first and last reason in this list are related to ease of use, which echoes the results from the interviews.

One slightly surprising result from the survey is how highly customers rated the importance of being able to do freehand drawing during a distributed meeting (mean of 4.83 on a 7 point scale where 1 = Not at all important and 7 = Very important). This was second only in importance to the ability to share web pages or browsers (mean of 5.41). Showing presentations (mean of 4.64) and sharing applications (mean of 4.30) were also important. Unfortunately we did not ask the question about importance of free-hand drawing in our internal study, so there is no way to directly compare the results, but we did learn that only 9.6% of internal people who used real-time collaboration tools used some sort of electronic whiteboard. This low frequency is perhaps explained by a mismatch in task. The free-hand drawing capabilities in a whiteboard application may not be adequate or appropriate for the sort of free-hand drawing our customers have in mind.

The customer survey has also provided us some insight into people's feelings about the effectiveness of the particular real-time collaboration tools they are currently using. On a 10 point scale where 1 = Very Poor and 10 = Excellent, the mean rating for tool effectiveness was 6.35. When Meeting Central is ready for pilot deployment, we plan to use this tool effectiveness measure as well as the meeting effectiveness measure from our internal study as baselines for comparison.

4. MEETING CENTRAL PROTOTYPE UI

We designed the Meeting Central collaboration suite to address as many as possible of the top distributed meeting problems that emerged from our internal study (recall that the software prototype was built prior to the customer-focused research). Since some of the problems relate to how people behave, they are difficult or impossible to solve with a technology solution. For example technology is not likely to turn chronically disorganized people into efficient advanced planners. Some behavioral issues, on the other hand, can be addressed by technology. For example, while it is helpful for people to say their name before they speak, this is an unnatural behavior, so easily forgotten in the heat of a discussion. Likewise, announcing the arrival of latecomers or the departure of people who must leave early is disruptive to the flow of a meeting. Technology can completely eliminate the need for these distributed meeting "best practices."

Speakers not checking for understanding in a distributed meeting is another problem that impacts meeting effectiveness. One of the reasons that a speaker does not often check for understanding is that it is extremely awkward to do this, particularly with many remote participants. A substantial chunk of every meeting could be eaten up if one took the time to ask each individual if they understand or have any questions. The issue of not checking for understanding is related to survey respondents' complaints about meetings not being well facilitated. While every meeting facilitator probably has room for improvement, we need to recognize that it is fundamentally difficult to facilitate a meeting in which you cannot see people's reactions, body language, or eye gaze. One of the goals of this project is to restore at least a small number of visual cues in order to help facilitators do their job more effectively.

Advanced planning is another area in which most people could improve, but some of the planning needed to ensure that a distributed meeting will run smoothly is a nuisance. For example, determining how to distribute materials created moments before

the meeting is always an issue. Setting up a mechanism to collect different people's contributions to a meeting is time-consuming. Technology cannot ensure that meeting facilitators have their thoughts well organized or their slides sensibly ordered, but we can at least help with the mechanics of providing a central repository of information for each meeting.

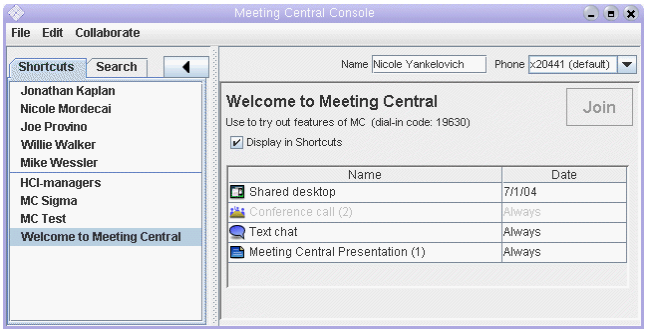
Audio problems were the single biggest factor in distributed meetings that were not effective. Some audio problems are difficult to fix, such as noisy long-distance phone lines, but most audio problems stem from quite simple sources. People are either making too much noise by shuffling papers near a microphone, breathing heavily, eating, etc., or they are attending the meeting from a noisy location such as their car, their home or a public place. Remembering to mute the phone when not speaking solves most of these problems. The other major source of audio problems is people not speaking close enough to a microphone or having a poorly adjusted telephone headset. With some minor effort, like passing a microphone to a person asking a question in a conference room, many of these problems can be solved. The bottom line is that most audio problems are, in fact, behavioral. They are compounded by the difficulty remote participants have, both technically and socially, in interrupting to indicate that the problem exists.

The design we came up with to address these problems is fairly minimalist. Our goal was to build a suite of lightweight tools that would be easy to use and contain only the features necessary to run an effective distributed meeting. We also wanted to avoid the problem of people not wanting to use the tool because the setup was too complex. Someone who uses Meeting Central once to join someone else's meeting should be able to do exactly the same steps to run his or her own meeting.

While it is easy to construct lots of extra controls for the speaker or facilitator, our philosophy was to let humans do what they are good at – mediating the social situation – and use technology to solve only those problems that people find difficult or impossible. The following sections describe the tools in the Meeting Central suite and illustrate how the features address each of the top distributed meeting problems.

4.1 Console

The Meeting Central Console is a Java application that users can leave running on their desktops. It provides the mechanism for searching for and organizing individual contacts and meetings. A shortcuts list provides quick access to recurring meetings and to people the user contacts most frequently.



The Console also provides a single place where meeting attendees can look for content associated with a meeting. When adding content to a meeting using the Console, users can specify if the document should always be made available or if it is intended for a particular date. Documents used on past dates can still be accessed through the Console, but they will not automatically open when the meeting is joined.

4.2 Telephony Features

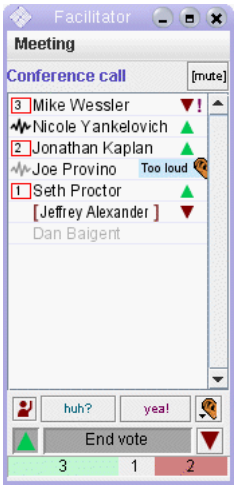
Most of Meeting Cental's telephony features are available through the Console's user interface. By default, the Console includes the user's default telephone extension in the Phone field (shown upper right). The user can type in a new phone number if they are not currently at this location, or they can select the software phone option to use their computer for the audio portion of the meeting rather than a standard telephone.

If the user needs to leave his or her current location during a meeting but wishes to stay on the call, it is possible to *migrate* the call to another phone. By selecting an alternate telephone number (or typing one in), the user can, for example, seamlessly continue the call on their cell phone without others in the meeting being aware of the transition. They can even quit out of the entire Meeting Central user interface while remaining on the call. If they have no graphical interface, they can instead use a minimal non-visual interface to perform some of the more essential functions of the Facilitator, described below.

If the user has no display available at the time of a meeting, they can access the non-visual interface directly from any telephone. We provide a single number that Meeting Central users can call into to identify themselves. When a meeting is set up, it is automatically assigned a unique code. Users calling into the main number enter this code to join their desired meeting. They will be identified by name to others who are attending the meeting using the graphical interface.

4.3 Facilitator

All meetings in Meeting Central, including those with only two participants, include an audio channel by default. When more than two people are involved in a meeting, the Facilitator can be used along with voice to show who is invited to a meeting (the grayed out names), who is present (the bold names), and who is speaking. A squiggle speaking indicator (shown before Nicole and Joe's names) appears next to the name of each person who is speaking, even if there is more than one.



To help remote participants indicate their desire to speak, the first icon in the tool bar at the bottom of the window allows users to raise their hands. When the hand raise button is pressed, an audio indicator alerts other meeting attendees that someone wishes to speak. In addition, a number is placed by each person's name to indicate the order in which hands were raised. The hand-raise indicators are only advisory; the software does not enforce this ordering in any way.

Several features help presenters or meeting facilitators check for understanding. Simple expression buttons (huh? and yea!) provide users with a back channel method of indicating they do not understand or that they agree with what is being said by placing either a question mark or exclamation point to the right of their name (see Mike's entry). In addition, an extremely simple voting

mechanism using up and down arrows allows meeting members to quickly indicate agreement or not. The meaning of the arrows can be mapped to whatever the person calling the vote wishes. For example, "Use the up arrow if you've finished reading the white paper and the down arrow if you need more time." Beneath the End Vote button is a dynamic tally of the current vote. For a small number of meeting participants this is not that useful, but in meetings with over 10 people, particularly if all the names do not fit in the window, the tally can provide a quick view of the outcome of the vote.

To address the issue of audio problems, the Facilitator provides what Erickson and Kellogg refer to as socially translucent features. These sort of features have the characteristics of visibility, awareness, and accountability [10]. An audio problem menu (ear icon to the right of the yea! button) allows people to indicate audio problems without verbally interrupting. For example, users can visually let people know if they are too loud or too soft (see Joe's entry), if there is too much background noise or if they have no audio at all. This raises awareness that a problem exists and makes the problem visible. Since it is much easier to use this menu than it is to verbally complain about an audio problem, people can use it more frequently, holding the people responsible for the problem accountable for trying to remedy it.

A mute button, in the upper right corner of the window, provides every user with the ability to mute their phone. When muted, a visual indicator appears to show the person's state (Jeffrey's audio is muted). Since the speaking indicator is triggered by loud noises as well as voice, the person making the noise is visible to everyone and they can be asked to use the mute facility.

If a user wishes to carry on a brief side conversation with someone else in the meeting, he or she can select the name of another user in the Facilitator window and select the "whisper" menu command. This automatically mutes the person's audio to the meeting and opens a separate whisper group window with a "Push to whisper" button. While the button is depressed, audio is transmitted only to the selected person. A speaking indicator appears in the whisper group window to provide concrete feedback as to who can hear the whisper. The person whispering continues to hear the audio from the main meeting, but at a lowered volume level.

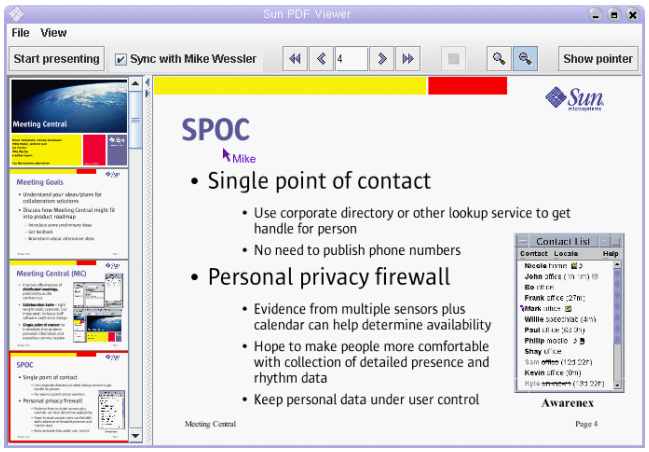
As mentioned in the "Telephony Features" section above, users can migrate their phone call such that they may no longer have a graphical interface. In this case, they can use the telephone keypad to mute and unmute their phone, raise and lower their hand, vote, and request the name of the person who is currently speaking.

While the non-visual interface is handy, it is difficult to remember the keypad commands. To help address this, we have designed the Facilitator to work on mobile platforms as well as on desktop and laptop computers. We have recently started a project to port the software to a PDA and also hope to demonstrate it working on a cell phone in the future.

4.4 PDF Viewer

Since sharing presentations is one of the most important activities in distributed meetings, the first data sharing tool we have prototyped in our suite is designed specifically for this task.

The PDF Viewer is a multi-user tool for sharing PDF documents. Anyone in a meeting can open and share a PDF document. To become the presenter, a user simply has to click on the "Start Presenting" button. A control next to this button allows other attendees to synchronize or unsynchronize with the presenter at any time. By default, the view is synchronized if the PDF Viewer is launched while someone is presenting. This allows late-comers to know exactly which slide the presenter is on and it also eliminates the need for presenters to speak the slide numbers.

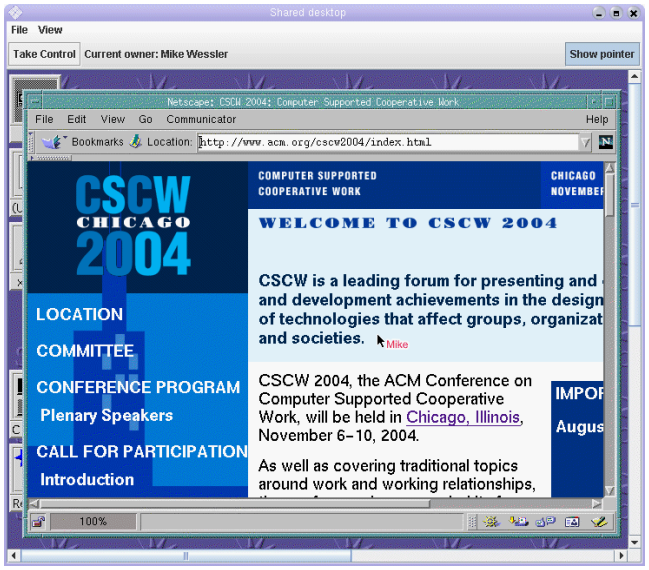


The presenter or any meeting attendee can opt to turn on his or her *telepointer* using the "Show pointer" button. This allows meeting participants to point and gesture in the context of the presentation.

Unlike webcasting technology or even general-purpose application sharing tools, the PDF Viewer is a collaboration-aware application in the sense that each user has his or her own copy of the PDF document. This allows users to browse the document independently and re-synchronize with the presenter at any time. With the other technologies, there is one central view of a document being shared which forces participants to be in lock-step with the presenter.

4.5 Shared Desktop

We added a Shared Desktop to Meeting Central after conducting our customer research where we discovered that sharing web pages was number one on customers' list of important activities and sharing applications was fourth on the list.



This application uses a standard VNC server [17] with a special viewer that has been enhanced with floor control and telepointers. Like the PDF Viewer, the Shared Desktop allows any meeting participant to "Take Control." The major difference is that this application is not truly collaboration-aware. All people viewing the Shared Desktop must be in lock step with one another.

In the future, we plan to add a collaboration-aware web browser that will allow users to browse pages independently or synchronize with a presenter. We also hope to encourage developers to add collaboration-aware versions of other commonly used applications. Since it is unrealistic to expect collaboration-aware versions of every application users may wish to share, the Shared Desktop will continue to provide this capability for applications that have not been specially tailored for multi-user use.

4.6 Addressing the Problems

The following list summarizes how the Meeting Central prototype addresses the distributed meeting problems uncovered in our internal study.

How Meeting Central Addresses Top Problems

Audio Solutions
Speaking indicator makes source of extraneous noise visible
Audio problem menu makes audio problems visible
Visibility helps make people accountable for resolving problems
Audio problem menu does not cause verbal interruptions
Background noise lessened by providing mute for everyone
Mute button provides awareness of state

Behavior Solutions
Need for hard-to-recall behaviors (e.g., saying name) eliminated
Need to verbally specify slide/page numbers eliminated
Hand raising makes people's desire to speak visible
Voting allows speakers to check for understanding
Adv. planning less necessary: docs can be added during meeting

Technical Solutions
Documents are in context of meeting, not in separate application
Meeting as repository makes documents easy to locate
Tool necessary to view document is downloaded with document
Shared Desktop supports sharing of any type of document
Speaking indicator makes visible who is speaking
Who is in meeting and who has joined or left also visible

4.7 Preliminary User Reactions

Meeting Central is still a work in progress. The software continues to evolve every day. We have so far gathered a small amount of usage data. Beginning in March 2004, three groups ranging in size from 4 to 15 people have started using Meeting Central for their weekly meetings. Our development team has also used the software to brief senior executives about our project.

From this limited experience, we have received the most favorable comments about the speaking indicator. One user commented, "the integration of voice conferencing features is very rich, particularly automatically denoting which participant is speaking with a 'presence'-like icon in the Facilitator window." Some meetings have made extensive use of the handraising feature. "I think this thing is really helpful," another user said, "you can see exactly who wants to speak." People have also commented positively on the visibility of mute. One person said, "We caught someone trying to talk on mute." The slide synchronization feature in the PDF Viewer is another well-liked feature. "It's nice not to have to worry about what slide the speaker is on," a user commented. "If I get distracted for a minute, I don't have to worry about being lost."

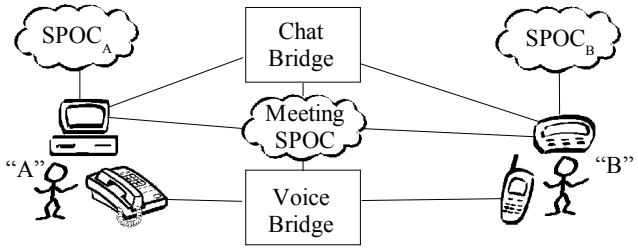
Some of the additional features these early users have requested are in our project plan, such as better integration with common calendar applications, quicker access to meetings through web links, and more collaboration-aware applications. Those people who have tried to put affinity style brainstorming into practice in their projects have clamored particularly loudly for a tool that supports a distributed method for doing this sort of brainstorming. Other feature requests have included the addition of video conferencing, multiple choice voting options, and a wider range of buttons to express emotions such as smiling, blushing and clapping.

Based on this feedback and some user testing of individual features, the team has iterated several times on the design of the implemented tools. In the near future we hope to be able to deploy the software widely enough to gather data about the effectiveness of the overall tool suite and the impact it will have on meeting effectiveness compared to our baseline measures.

5. ARCHITECTURE

The central design concept for the Meeting Central architecture is the SPOC or "single point of contact."

As shown in the following diagram, each person in the Meeting Central system has a SPOC. Each person typically runs a client that communicates with their SPOC and other people's SPOCs. For example, the Meeting Central Console is such a client.



All communication with others is done via a Meeting. The Meeting is a specialized SPOC designed to support collaborations. To communicate with others, a person will use their client either to create a meeting "on the fly," or to join an existing meeting. Once in a meeting, people can simply invite others to collaborate with them in that meeting.

The SPOC is supported by a framework that provides a distributed system that is secure, extensible, scalable, open, and easily integrated with existing communications tools. By making the design decision that all communications are done via meetings, the SPOC framework easily permits ad-hoc discussions between two people to turn into meetings of three and more.

The SPOC infrastructure also permits easy integration with various communications tools. For example, the SPOC infrastructure allows people to use existing instant messaging systems such as those based on the Extensible Messaging and Presence Protocol (XMPP) [20], and also provides sophisticated integration with both public-switched-telephone network (PSTN) and voice-over-IP (VoIP) telephony systems.

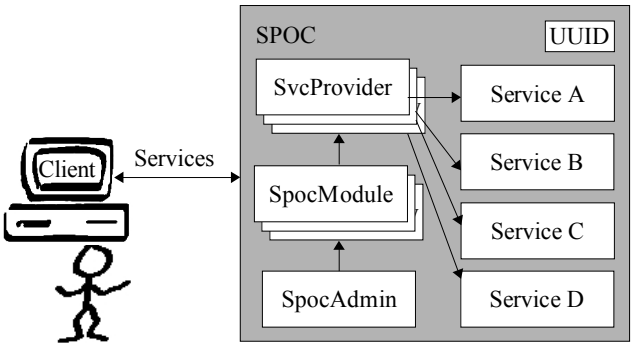
5.1 SPOC

The SPOC class is a general-purpose secure network service container. SPOCs need not live on the same machine. In fact, we expect the system to be quite decentralized. Clients on the network discover a SPOC using Jini™ technology [3] and begin

interacting with the SPOC via its services. For example, a client may discover the SPOC for a person and begin interacting with its availability service.

To manage security, the SPOC framework requires both the SPOC and the client accessing the SPOC to mutually authenticate with each other using Transport Layer Security (TLS) [9]. With TLS, the clients and SPOC not only mutually authenticate with each other, but also establish an encrypted communication channel between them.

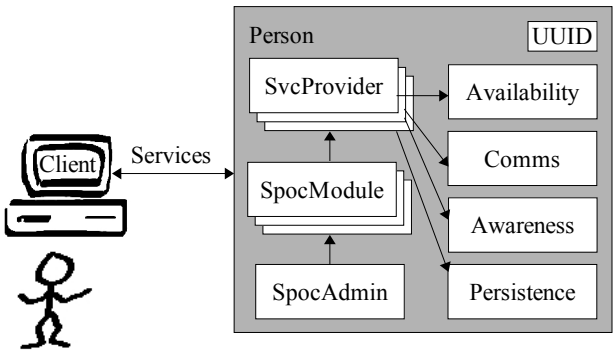
To manage privacy, the SPOC framework makes use of the eXtensible Access Control Markup Language (XACML) [16] system, which is a generic, flexible, policy-based access control mechanism. We use XACML to determine both if clients can access services of a SPOC, and if clients are allowed to call specific methods on the services in the SPOC.



Given the richness of the Java™ programming language, the services of a SPOC are not fixed. With the proper authorization, a client may dynamically add, change, or remove services in a SPOC via a specialized SpocAdmin service. As such, the SPOC provides a generic, extensible, and secure mechanism for grouping and maintaining access to a set of services.

5.2 Person

The Person is a specialized SPOC that represents a human being. From a programming point of view, the only specialization provided by a Person class over a SPOC class is that the Person provides the notion of some form of human readable identity such as the person's name.



In practice, the Person SPOC maintains a dynamic set of services that are specific to an individual. For example, an availability service determines if the user is available for collaboration, a communications service provides communications endpoints like voice and IM, and an awareness service provides additional information such as presence and calendar details. As with any SPOC, the Person may also include a persistence service to provide backup in the case of a system failure.

Interaction with the services is managed via the SPOC security mechanism: clients are only allowed to access services to which they are authorized. Furthermore, once clients obtain access to a service, the service will only allow them to perform authorized operations on that service. For example, depending upon the identity and credentials of the client accessing the SPOC, the client may not be able to see complete awareness information about a user, but may only be able ask for a user's availability, which is partially derived from the awareness service. In the same vein, some special users, like the owner of the SPOC, may be able to update awareness information, whereas others may be denied access.

With this system, a Person SPOC maintains a secure environment to the "outside world," and only permits access to the information that the owner of the SPOC authorizes.

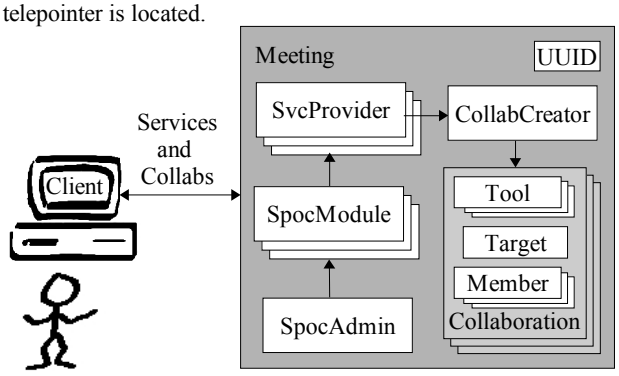
5.3 Meeting and Collaboration

As mentioned, a Meeting is another specialized form of the SPOC. The Meeting significantly extends the SPOC to include the notion of a c*ollaboration*.

As with real life meetings, people meet to collaborate about one or more things. Our notion of collaborations include tangible things such as PDF documents, presentations, and shared whiteboards, but also include more intangible things such as voice conferences and chat rooms.

Each collaboration in a Meeting comprises three things: the target, the members of the collaboration, and the set of end-user tools to interact with the collaboration.

To help illustrate the concept of a Meeting Central collaboration, consider a PDF document. With Meeting Central, the PDF document itself becomes the *target* of the collaboration. For clients that do not have a collaboration-aware PDF viewer available, they can simply download a *tool* from the collaboration. Once a person joins a collaboration, they become a *member* and the collaboration maintains the interaction between all the members of the collaboration. In the PDF example, the collaboration will maintain information about who is presenting, which page each member is on, and where each member's telepointer is located.



With this flexible system, users can dynamically add new target types to the system without needing to upgrade their clients or requiring system administrators to "reboot" the system. Furthermore, the security mechanism of the SPOC allows meetings to be as open or as closed as desired.

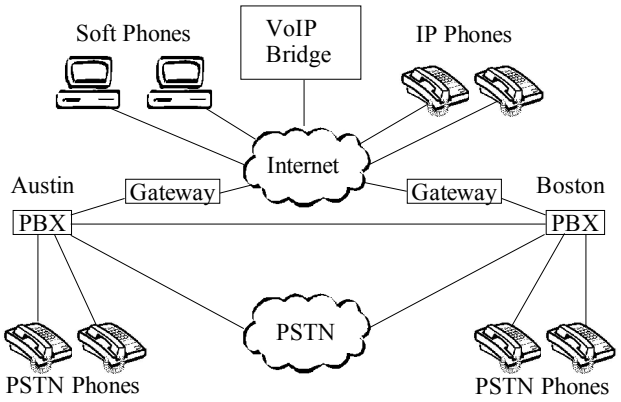
Worthy of mention, but not illustrated, is the ability to record the history of a meeting. For example, the meeting can record the

audio for an entire meeting, annotating who spoke when. This recording capability can be extended to other significant events such as slides changes, the results of votes, and so on.

5.4 Software Conference Bridge

To help with the level of telephony integration we desired, it was necessary to create a software conferencing bridge. The bridge is written entirely in the Java programming language, and performs all of its work using voice-over-IP (VoIP).

The software conferencing bridge, labeled "VoIP Bridge" in the drawing below, accepts commands from meetings to create voice conference rooms on the fly and uses the Session Initiation Protocol (SIP) [18] to call people into the conference.
The bridge can operate solely in a VoIP environment to connect standalone hardware IP phones and software IP phones, ones that use software to give a headphone/microphone attached to a computer telephone-like behavior.



The bridge also supports a VoIP-to-PSTN (Public Switch Telephone Network) gateway to call people with "landline" phones. This configuration provides a simple migration path for people to move from a PSTN environment to a VoIP environment without having to switch entirely to VoIP in one fell swoop.

By using VoIP, the bridge is able to assign a single audio channel for each participant in a conference. Having one channel per person permits a wide variety of features that would have been quite difficult to do with traditional telephony systems. For example, the bridge has the following capabilities: play customized audio treatments to one or more participants, raise or lower (and even mute) individual volume levels, detect DTMF key presses, determine who is present, record audio, and perform audio analysis such as identifying who spoke when. Audio recordings can be played back for future review, and potentially sent to a manual or automatic transcription service to be converted to text. Furthermore, since the bridge calls people rather than the other way around, users do not need to know phone numbers and passcodes to join a meeting, nor do they need to know phone numbers to contact individuals. Finally, because the software bridge can run entirely on a system inside the enterprise, businesses can achieve significant cost savings over traditional conference call hosting services.

6. CONCLUSION AND NEXT STEPS

Working with developers outside our team to validate the usefulness of the SPOC architecture as a general-purpose collaboration framework will be one of the immediate next steps in the Meeting Central project.

On the user interface side, we plan to implement several additional tools and features, prioritized based on the user research with customers. As we learned, joint web browsing and free-hand drawing are two priorities that our software does not currently support. Another issue, in particular, that we hope to address in the future is taking conference rooms into account. Both our internal and customer data clearly shows that people are still meeting face-to-face with at least those local to them. An effective distributed meeting tool must gracefully accommodate meeting configurations with a combination of conference rooms and individual remote attendees.

Although there is more functionality we plan to add, Meeting Cental's current capabilities address the majority of top problems people have with distributed meetings. In particular, the software addresses a number of behavior problems that, in the past, have been addressed at our company primarily through employee training. The need for many of the distributed meeting best practices that the training encourages are eliminated by features that make Meeting Central socially translucent. The system provides *visibility* and *awareness* of who is in a meeting, who is speaking, who has joined or left, who is presenting, who is on mute, who is voting, and who is having audio problems. As Erickson and Kellogg point out, this type of visibility and awareness "brings our social rules into play" [10], creating an environment where individuals feel *accountable* for resolving problems. "In socially translucent systems," they write, "we believe it will be easier for users to carry on coherent discussions."

With Meeting Cental's minimalist approach to collaboration tools, we have streamlined the feature set to include those with the most impact on distributed meeting effectiveness.

7. REFERENCES

[1] Ackerman, Mark S., Starr, Brian, Hindus. Debby, Mainwaring, Scott D. Hanging on the 'wire': a field study of an audio-only media space. ACM Transactions on Computer-Human Interaction (TOCHI), 4, 1 (March 1997), 39-66.

[2] Ahuja, S. R., J. Ensor, Robert, Horn, David N. The Rapport multimedia conferencing system. ACM SIGOIS and IEEECS TC-OA Conference on Office information systems, *ACM SIGOIS Bulletin*, 9, 2-3 (April 1988).

[3] Arnold, K., O'Sullivan, B., Scheifler, R.W., Waldo, J., Wollrath, A., *The Jini Specification,* Addison-Wesley, Reading, MA, 1999, ISBN 0-201-61634-3.

[4] Begole, James "Bo", Rosen, Mary Beth and Shaffer, Clifford A. Flexible collaboration transparency: supporting worker independence in replicated application-sharing systems," *ACM Transactions on Computer-Human Interaction,* 6, 2 (June 1999), 95-132.

[5] Bentley, R. and Appelt, W., Designing a system for cooperative work on the World Wide Web: Experiences with the BSCW system. In *Proceedings of the 30th Hawaii International Conference on System Sciences* (Maui, Hawaii, USA, January 7-10, 1997). IEEE Computer Society Press, 1997.

[6] Cadiz, JJ; Narin, Attila; Jancke, Gavin; and Gupta, Anoop. Exploring PC-Telephone Convergence with the Enhanced Telephone Prototype. In *Proceedings of the SIGCHI*

conference on Human factors in computing systems (CHI 2004) (Viena, Austria, April 24-29, 2004). ACM Press, New York, NY, 2004, 215-222.

[7] Colburn, Alex R, Cohen, Michael F., Drucker, Steven M., LeeTiernan, Scott, and Gupta, Anoop. *Graphical Enhancements for Voice Only Conference Calls.* Technical Report MSR-TR-2001-95, Microsoft Research, October 1, 2001. http://research.microsoft.com/research/pubs.

[8] Cook, Peter, Ellis, Clarence, Graf, Mike, Rein, Gail, and Smith. Project Nick: Meetings augmentation and analysis. *ACM Transactions on Information Systems (TOIS),* 5, 2 (April 1987), 132 – 146.

[9] Dierks, T., Allen, C., *The TLS Protocol Version 1.0.* (January 1999), IETF RFC 2246, http://www.ietf.org/rfc/rfc2246.txt.

[10] Erickson, Thomas and Kellogg, Wendy A. Social translucence: an approach to designing systems that support social processes. *ACM Transactions on Computer-Human Interaction,* 7, 1 (March 2000), 59-83.

[11] Geyer, Werner, Richter, Heather, Fuchs, Ludwin, Frauenhofer, Tom, Daijavad, Shahrokh, and Poltrock, Steven. Team collaboration space supporting capture and access of virtual meetings. In *Proceedings of the 2001 International ACM SIGGROUP Conference on Supporting Group Work (Group 2001)* (Boulder, Colorado, USA, September 30 – October 3, 2001) ACM Press, New York, NY, 2001, 188 – 196.

[12] Greenberg, Saul. Enhancing creativity with groupware toolkits. Invited keynote talk in *Proceedings of the 9th International Workshop on Groupware (CRIWG 2003)* (Autrans, France, September 28 - October 2, 2003). Lecture Notes in Computer Science (LNCS 2806), Springer-Verlag, Berlin, Heidelberg, New York, 2003, 1-9.

[13] Isaacs, Ellen A., Morris, Trevor, and Rodriguez, Thomas K. A Forum for Supporting Interactive Presentations to Distributed Audiences. *Proceedings of the Conference on Computer-Supported Cooperative Work (CSCW)* (Chapel Hill, NC, October 22-26, 1994) ACM Press, New York, NY, 405-416.

[14] Isenhour, Philip, Begole, James "Bo", Heagy, Winfield S., and Shaffer, Clifford A. Sieve: a Java-based collaborative visualization environment. In *Proceedings of the 8th IEEE Visualization '97 Conference* (Phoenix, AZ, USA, October 22-24, 1997). 1997, 13-16.

[15] Moors, Tim: The SmartPhone: Interactive group audio with complementary symbolic control. In *Proceedings of the 4th international conference on Distributed communities on the web (DCW 2002)* (Sydney, Australia, April 3-5 2002). Lecture Notes in Computer Science (LNCS 2468), Springer-Verlag, Berlin, Heidelberg, New York, 2002, 100-112.

[16] OASIS Committee. *The eXtensible Access Control Markup Language (XACML) Version 1.1* (August 2003), http://www.oasis-open.org/committees/xacml/repository/cs-xacml- specification-1.1.pdf.

[17] Richardson, T.; Stafford-Fraser, Q.; Wood, K.R.; Hopper, A.; Virtual network computing. In *Internet Computing*, IEEE, Volume: 2, Issue: 1, Jan.-Feb. 1998, 33 – 38.

[18] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and Schooler, E., *SIP: The Session Initiation Protocol* (June 2002) IETF RFC 3261, http://www.ietf.org/rfc/rfc3261.txt.

[19] Roseman, M. and Greenberg, S., TeamRooms: network places for collaboration, In *Proceedings of the 1996 ACM conference on Computer supported cooperative work (CSCW 96)* (Boston, Massachusetts, USA, November 16-20, 1996). ACM Press, New York, NY, 1996, 325-333.

[20] Saint-Andre, P., *Extensible Messaging and Presence Protocol (XMPP): Core.* January 2004, http://www.ietf.org/internet-drafts/draft-ietf-xmpp-core-22.txt.

[21] Tang, John C., Yankelovich, Nicole, Begole, James "Bo", Van Kleek, Max, Li, Francis, and Bhalodia, Janak. ConNexus to Awarenex: extending awareness to mobile users. In *Proceedings of the SIGCHI conference on Human factors in computing systems (CHI 2001)* (Seattle, Washington, USA, March 31 – April 5, 2001). ACM Press, New York, NY, 2001, 221-228.