Publications Saul Greenberg GroupLab Dept Computer Science University of Calgary

Grouplab at SkiGraph

Michael Boyle, Shaun Kaasten, Michael Rounding, James Tam, Ana Zanella (students)
Saul Greenberg, Sheelagh Carpendale and Frank Maurer(supervisors)
Department of Computer Science,
University of Calgary
Calgary, Alberta Canada T2N 1N4
+1-403-220-6087
saul@cpsc.ucalgary.ca

Cite as:

> Boyle, M., Kaasten, S., Rounding, M., Tam, J. and Zanella (Students); Greenberg, S., Carpendale, S. and Maurer, F. (Supervisors). **Grouplab at Skigraph,** Report 2000-652-04, Department of Computer Science, University of Calgary, Calgary, Alberta, Canada. March.

The Western Computer Graphics Symposium, nicknamed 'SkiGraph', is an annual professional meeting comprising mostly graphics researchers and their graduate students from Western Canada. In 2000, several Western Canadian researchers in Human Computer Interaction: Saul Greenberg (U. Calgary), Carl Gutwin (U. Saskatchewan), Kori Inkpen (Simon Fraser) and Sheelagh Carpendale (U. Calgary) agreed to use SkiGraph as a way to get themselves and their graduate students together, where students would present papers describing their research. Because it was important for all graduate students to share their ideas, the papers written could range from identification of research areas and tentative proposals of research problems all the way to detailed results from mature work.

This research report collects five research papers by students at Grouplab to SkiGraph (Grouplab is the laboratory for human computer interaction research at the University of Calgary). The papers are listed below. In all cases, the students are the first author followed by faculty members who have supervised or contributed to the work in one way or another. Individual papers may be cited directly by including the following information:

> In *Proceedings of the Western Computer Graphics Symposium 2000*, (Panorama Mountain Village, BC, Canada), March 26-29, 2000.

The Papers

Balancing Awareness and Privacy in a Video Media Space Using Distortion Filtration	Michael Boyle Saul Greenberg	PDF
Designing an Integrated Bookmark / History System for Web Browsing	Shaun Kaasten Saul Greenberg	PDF
Awareness Projected: Moving Awareness to a Public Space	Michael Rounding Saul Greenberg Sheelagh Carpendale.	PDF
Change Management.	James Tam Saul Greenberg Frank Maurer	PDF
A Single Display Groupware Widget Set	Ana Zanella Saul Greenberg	PDF

Balancing Awareness and Privacy in a Video Media Space Using Distortion Filtration

Michael Boyle
University of Calgary
2500 University Drive NW
Calgary AB T2N 1N4 CANADA
+1 403 220-3532
boylem@cpsc.ucalgary.ca

Saul Greenberg
University of Calgary
2500 University Drive NW
Calgary AB T2N 1N4 CANADA
+1 403 220-6087
saul@cpsc.ucalgary.ca

ABSTRACT

Collaboration among distributed workgroup members is hampered by the lack of good tools to support informal interactions. These tools either fail to provide teleawareness or enable smooth transitions into and out of informal interactions. Video media spaces—always-on video links—have been proposed as a solution to this problem. However, the “always-on” nature of video media spaces results in a conflict between the desire to provide awareness and the need to preserve privacy. The present study examines distortion filtration applied to always-on video as means of resolving this tension. Our discussions include the inter-related concepts of informal interactions, awareness, and privacy; and the treatment afforded by existing distributed collaboration support tools. We then outline the present study, where our goal is to understand the effect of distortion filtration on awareness and privacy.

Keywords

Tele-awareness, telepresence, privacy, informal interaction, video media spaces, distortion filtration, distributed collaboration.

1. INTRODUCTION

Collaboration may be loosely defined as people working together towards common goals, sharing ideas, and exchanging information. Humans have a flexible protocol for managing collaborative interactions that facilitates smooth and efficient collaboration, that is, collaboration in which content is communicated freely, correctly, and with a minimum of hassle. This protocol relies on a combination of subtle and explicit signals for expression including words, vocal intonations, gaze, posture, and gestures. Because much of this protocol was developed between people who are physically proximate to one another, it is no surprise that groups generally collaborate best when their members are co-located.

Instead of enjoying unfettered face-to-face contact—arguably the most natural way for humans to interact—distributed workgroups must rely on telecommunication media and computer-supported collaboration tools to work together. Cognitive psychologists suggest that for such a tool to effectively support collaborative activities, it must enable people to utilize the same social protocol for governing interaction as that which is used for face-to-face interactions [7]. While a plethora of tools for distributed collaboration has been developed, very few are capable of supporting people’s natural social protocol.

In this paper, we will consider the design of a video-based media space that supports the way people naturally balance privacy and awareness. We begin in Section 2 by describing what we mean by informal interaction, and will include a summary of how technology tries to support this. In Section 3, we discuss how awareness of others is fundamental to supporting informal interaction, followed by how media spaces have been designed to support the two. Privacy concerns are presented in Section 4. Sections 5 and 6 describe our current approach. We introduce filtration as a means for balancing awareness and privacy, we discuss lessons learned by other researchers as well as ourselves, and we detail the study of the effects of filtration on awareness that we are currently running.

2. INFORMAL INTERACTIONS

2.1 Nature and value

At the heart of any collaborative process lie informal interactions. These are brief and spontaneous interpersonal exchanges where the participants, location, timing, and even agenda are not planned in advance [9]. Transitions into and out of informal interactions are made implicitly, and are thus the process is graceful, lightweight, and recoverable. Similarly, changes within informal interactions may also be made implicitly, making informal interactions smooth and adaptive. Lastly, in physically co-located situations, we find that informal interactions capitalize on a wide array of real-time communication channels (e.g., visual, aural, sensory), and thus are rich, interactive, and adaptive.

The unplanned nature of informal interactions is important. Because they are lightweight enough to allow participants to capitalize on serendipity and spontaneity, informal interactions occur very frequently. Because participants do not invest much energy in forethought and preplanning, they are free to adapt the nature of their interactions to best fit their immediate needs. Indeed, it has been claimed that informal interactions would not occur if they had to be planned [9]. In practice, the flexibility of informal interactions means they may be easily abandoned or postponed, that they can quickly shift between social and work contexts and between private and public matters, and that episodes of these interactions can effortlessly gain or lose participants. The length can be a few seconds but can easily stretch into hours. This level of freedom and degree of control over the particulars of interaction is rarely found in pre-planned interactions, which typically do not deviate much from their prescribed formats and agendas and, as a result, are often seen as slow and incomplete.

Although informal in tone, interactions of this type are of great importance to collaboration. Much valuable work-oriented information is exchanged by way of it. Furthermore, much of this information is not exchanged by other means [9]. Thus many experts in groupware argue that a distributed collaboration environment's failure to support informal interactions will be to the detriment of its users.

2.2 Example systems supporting informal interactions

Communication tools designed to support informal interactions include examples such as the telephone, desktop videoconferencing (DVC), and experimental systems such as CRUISER [4,11]. These tools all follow a connect-interact-disconnect engagement model. This model works best for planned interactions and is advantageous because it reserves the highest quality of service for times of actual interaction—when it is needed most. However, this model isolates distributed group members when not directly engaged in interaction, and makes it harder for them to recognize opportunities for informal interaction.

Placing a call is heavyweight and requires explicit thought and action. Telephones mitigate this problem through ubiquity and fast connect times. DVC systems, where connections typically take 8 to 15 seconds to establish, cannot claim such an advantage. Worse, the incremental benefit to the outcome of collaborative tasks attained by adding video to an audio-only conference has been shown to be small [12,13,14]. This is not to say that video is without merit: there is strong evidence that adding video increases users' satisfaction with the collaborative process and the communication media.

The CRUISER system offered random connections to support serendipitous informal interactions (like bumping into a colleague in on one's way to the printer), but in practice was found to end up connecting strangers who had no interested talking with one another. Although serendipitous informal interactions are still possible in systems that use the telephone-model for engagement (e.g., when someone other than the intended person answers the call, or is present when the call takes place), the process is complicated by the fact that callers cannot tell if it is a good time to call unless they place the call and explicitly ask. Aside from being explicit, systems that use the telephone engagement model are intrusive. For example, in interviews conducted during observational studies of the CRUISER system, users reported that the video windows—which just popped up unexpectedly on their displays—were at times quite disruptive, much like the loud ring of an unexpected telephone call.

2.3 Summary

The key weakness found in all of the collaboration tools mentioned here is an inability to provide the teleawareness and sense of telepresence that are needed for informal interaction. They fail to provide enough informal awareness to infer presence and availability, or they do so in an awkward way. Next, we address teleawareness and the role it plays in supporting informal interactions.

3. AWARENESS FOR INFORMAL INTERACTIONS

3.1 Nature and value

Awareness is an unobtrusive background understanding of our shared work environment, the people in it, and their activities [7]. People incessantly accrue awareness information from subtle and implicit visual and auditory cues sensed and tabulated at the periphery of our consciousness. Rarely does one actively seek to acquire awareness.

Informal awareness is a sense of the presence, availability, and activities of others. It is particularly easy to acquire this kind of awareness when workgroup members are physically proximal to one another [7,9].

Conversational awareness is an understanding of the state of one's conversation. It is picked up from visual cues, such as eye contact, facial expressions, gestures and posture, and from auditory cues such as intonation and the use of particular words [7]. Conversational awareness is tightly coupled with informal interactions, and demands use of very rich and high quality channels such as those provided by face-to-face interactions.

Both informal and conversational awareness are critical to establishing and regulating informal interactions and facilitates transitions into, out of, and within informal interactions in many key ways. Informal awareness helps people stay on top of who is available or interruptible. Indeed, before interaction is engaged, this awareness helps participants decide many of the particulars of interaction. Knowing who is present or available helps one decide whom to contact, and this, in turn has implications for the choice of medium, locale, and timing. Conversational awareness is the bridge that lets people act upon informal awareness, turning opportunity into conversation.

Conversational awareness makes interactions smoother by helping to regulate the flow of interaction. It helps speakers and listeners exchange answers to mechanical and affective questions, so as to make incremental refinements to interaction [7]. By constantly incorporating small changes derived from conversational awareness cues, changes within the conversation—for example, speaker/listener role reversals, and shifts in topic—can be made in a less abrupt and less explicit fashion.

Awareness is not a one-sided affair: all participants maintain some awareness of all of these aspects. This mutual understanding facilitates co-operative decision-making and allows participants in informal interactions to quickly decide such things as: the entry or departure of a participants; changes of locale; and, whether to postpone further discussion and reschedule it for a better time.

3.2 Example systems supporting informal awareness

Teleawareness provided by existing support systems vary widely in fidelity. At the coarse end of the spectrum are instant messengers (IM) such as ICQ [8] and MSN Messenger [10]. These systems couple synchronous chat and lightweight asynchronous messaging with on-line presence information. One's on-line state is shown using an iconic representation, with states such as available, busy, away, and off-line. In practical experiences, even this crude sense of telepresence seems to be better than none, but there remains a need to further examine patterns of instant messenger use before firm conclusions may be drawn.

Portholes [3] and similar systems lie in the middle of the teleawareness information spectrum. These provide periodic video snapshots from a federation of remote sites. This low-bandwidth approach to providing long-term disengaged teleawareness generated positive feedback from users in field studies. It enriched the sense of community both remotely across sites, and locally within each site.

At the high end of this spectrum are video media spaces: always-on video links, typically among offices or common spaces [1,5,13,14,15]. The fact that the audio and video channels are always left on is critical to the potential success of video media spaces. It allows users to employ the same social protocol governing co-located interactions to managing interactions in a distributed setting. Furthermore, it enables participants to maintain mutual awareness of each other during engagement transitions, and so these transitions are made smooth, graceful, and recoverable. Experiences with video media spaces [1,5,13,14] show that by supporting such lightweight and natural transitions into and out of engagement, they extend the effective interaction space.

3.3 Summary

Just as there are many sources of teleawareness, so too are there substantially different strategies for providing teleawareness. These differ in fidelity from instant messengers to video media spaces. It remains to be seen if any of these tools actually provide the telepresence needed, but these approaches all appear promising in the few systems implemented. However, in video-mediated teleawareness systems such as Portholes and video media spaces, there exist obvious concerns regarding privacy [1,5,13]. Next, we address the issue of privacy and the treatment it has received in existing teleawareness and informal interaction support tools.

4. PRIVACY

4.1 Nature and value

Privacy is generally taken to be the ability to manage what information about oneself is made public, to whom it is made available, and for what purpose it is used [7]. The ability to preserve one's privacy is often taken together with the right to ensure that any information about oneself is accurate and used fairly.

Solitude is a related concept often mistaken for privacy. Solitude is the ability to manage interruptions and distractions [7]. Since "interruptions" and "distractions" are felt to be negative concepts, it may be better to think of solitude as the ability to manage the "if, when, why, how and with whom" of engaging in interaction.

The preservation of privacy in co-located settings hinges on reciprocity in our collaborative environment [7]. Reciprocity is a simple rule regarding interaction: in order to see, one must be able to be seen, or in order to hear, one must be able to be heard. Reciprocity helps mitigate privacy concerns by discouraging eavesdropping or spying, by making it difficult to get away with such behaviors, and by letting people know that they are being watched so they can adjust their behaviors accordingly.

Awareness in co-located settings is reciprocal, and mutual awareness helps us preserve each other's solitude. When participants are aware of the availability of one another, they are better able to make the decision whether to engage in interaction. It should be clear that a distributed collaboration support tool that forces a user to turn it off in order to preserve any measure of privacy or solitude (like video media spaces or telephones) is poorly designed. Turning the system off defeats the system's purpose because it no longer allows any awareness to flow through.

4.2 Revisiting systems by privacy

Although widely known to be insecure, telephones are generally considered a comfortable tool for exchanging sensitive information because of its predominately dyadic character. This seems to defy the fact that it is known to be insecure, and may be the result of acclamation over long-term use. Furthermore, the telephone is also quite inadequate at preserving one's solitude, because it fails to provide availability awareness: the only way to know for sure if someone is available for interaction by telephone is to call him and interrupt him, and therefore invade his solitude. Screening calls is not entirely satisfactory because it cuts off all means to quickly pass urgent messages that supersede the need for solitude, and confuses the "absent" and "unavailable" states. As it was based on the telephone model for engaging in interaction, the CRUISER system suffered from similar problems related to intrusiveness. Surprisingly, most users were comfortable with the idea of strangers looking into their offices, and this is likely because the system enforced reciprocity.

Most instant messengers require username/password authentication, and use explicit (and typically reciprocal) authorization schemes for determining who will see one's on-line status and send one messages. Although IMs normally appear as unobtrusive icons, "nuisance" messages sent using them could be very obtrusive. As these kinds of frequent interruptions are possible in other communication media (e.g., telephone, face-to-face contact) it could be argued that such situations are best resolved by natural social protocol, such as by sending a curt "go away" message, or by making oneself more inaccessible.

While Portholes-like systems don't threaten solitude, they—along with video media spaces—have significant privacy issues. Privacy of outgoing content is a bigger concern than solitude violations [4] but most existing systems have only crude privacy support. There were facilities to edit the local snapshots taken by the Portholes system, but the interface implementation was hard to use and users easily confused the absence of fresh snapshots for "technical difficulties." Some media spaces support disconnected states, but, as when screening telephone calls or closing one's office door, such behaviors confuse unavailability with absence. Other media space designs ignore the privacy issue altogether, but observations of actual use show that even among intimate collaborators privacy-awareness tensions exist [1].

4.3 Summary

There has been, historically, poor support for preserving privacy and protecting solitude in distributed collaboration support tools. Privacy is itself a nebulous concept, highly dependent on context. Consequently, most designers have taken a lazy approach to the subject. While many raise it as a concern, few address it. This is a serious omission, for we expect that privacy over the Internet will be a key issue for the 21st century. Ideally, one would like to capitalize on a video media space's ability to provide awareness and lightweight transitions into and out of interaction, but balance these aspects against a real need to provide control over privacy and solitude.

5. DISTORTION FILTRATION

5.1 Nature and value

Distortion filtration is proposed as a possible means for striking a balance between the desire to provide teleawareness and the need to preserve privacy. It is the process of algorithmically manipulating the contents of the audio and video streams to selectively obscure various levels of detail in the sound or picture. The goal is to hide elements of the picture or sound that might be deemed sensitive —details which one wishes to keep private— while still giving remote participants the ability to extract just enough awareness and presence information to make confident decisions regarding one's availability and interruptability.

Many filtration effects are possible: pixelization; smoothing; convolutions (e.g., Sobel, Laplace operations); overlays; and, translatory distortions (e.g., wave, ripple, fish-eye) are but a handful. Several examples are illustrated in Figure 1. These can be also applied across a continuum of extents (i.e., levels), as illustrated in Figure 2. Furthermore, the filter may be coupled with an excitory "lens" which determines the level to filter at a given point in a given image in the sequence. Figure 3 gives two example excitory lenses. The activity-selective lens applies the filter if a pixel in the current frame differs substantially from the corresponding pixel in a reference frame—typically a snapshot of the scene background, taken when no one is present. The second example, a radial lens, applies the filter in ever increasing amounts outward from the center of the image.

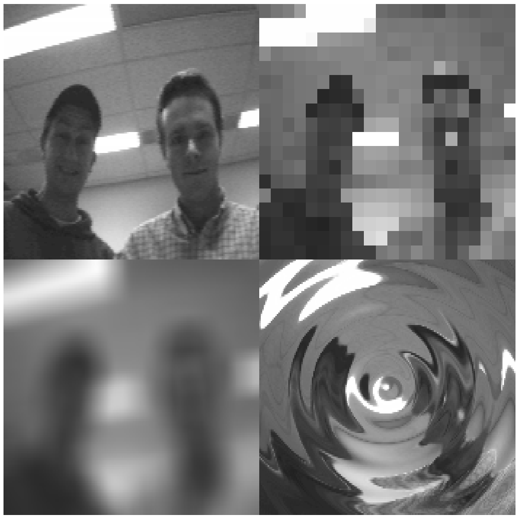


Figure 1. Example distortion filters: (clockwise from top-left) undistorted, pixelize, wave, and smoothing.

5.2 Existing systems

The visual channel, as suggested earlier, is an important part of interactions, but as demonstrated by the success of the telephone, it is by no means a requirement. Hudson and Smith [12] developed an audio media space (common audio channel shared among many participants) that does not disturb or annoy participants with unwanted audio fragments. Their report describes a system that provided both a background ambient audio signal, representing the overall audio activity level at a number of distant sites, as well as facilities to "tune in" to a specific speaker.

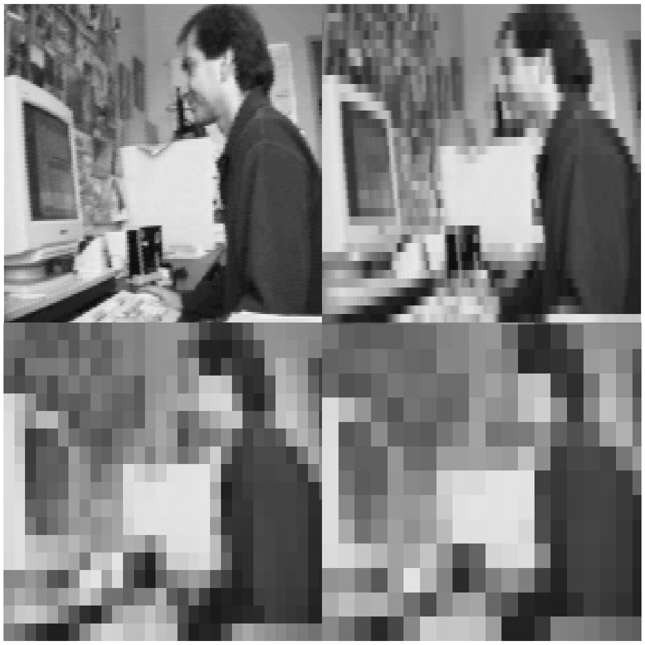


Figure 2. Pixelize filter at four different levels.

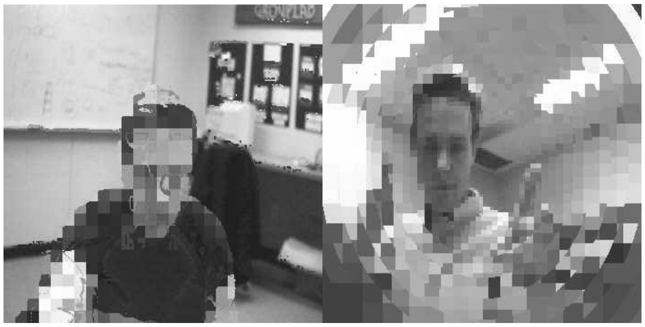


Figure 3. Pixelize filter used with two different excitory lenses: (left to right) activity selective, and radial.

Zhao and Stasko [15] applied a number of distortion filters to video clips, and then tested volunteers for accuracy in recognizing actors and activities. Filtration was performed at a single level. The filters examined were: pixelization; Sobel; a "shadow" filter (a pixelize filter coupled to a lens similar to the activity-selective lens discussed earlier, only that the filtration is applied to the reference frame); and a "live shadow" filter (an activity-selective lens). Twenty volunteers, mostly unfamiliar with the actors in the scenes shown, were used in tests. After viewing one of 21 video clips in one two sizes, a volunteer was asked specific questions about the location viewed, the actors in it, and their activities. The questions asked could be deemed "loaded", and the only metric recorded was the number of correct responses. Informal interviews were used to elicit qualitative information about usage and users' reactions, and the feedback given was incorporated into the design of an improved filter, the results of which have not yet been reported.

There are real-time performance issues when adding distortion filtration to live video links. A typical DVC application must capture, compress, transmit, receive, decompress and render video frames many times a second. Common desktop hardware/software/network configurations already have a difficult time keeping up with these core tasks, even at low frame rates, small frame sizes, and using highly lossy compression techniques. There is not much time left for sophisticated or computationally intensive filtering algorithms without the benefit of specialized hardware. Consequently, filtering algorithms toyed with thus far have been designed without much regard to human physiology and perception. Moreover, even if one is lucky enough to get an algorithm to work within real-time constraints, there does not yet exist a systematic way of testing a filter for its effects on awareness and its effectiveness in preserving privacy.

5.3 Summary

The idea of using distortion filtration in a video media space seems, at least superficially, like a good approach to balancing the tension that exists between maintaining awareness and preserving privacy. However, numerous questions remain unresolved by the work already done in this area. What is the impact of a filter on one's ability to extract awareness cues from a video scene? What are the characteristics of an effective privacy-preserving distortion filter? To what range of extents should a particular filter be applied? How should the extent of filtration across its range of useful values be controlled?

6. PRESENT STUDY

We are in the process of constructing always-on video links for awareness and interaction that reflect the lessons learned from previous experience. To this end, the present study will examine the impact the pixelizing and smoothing filters have at each of ten different extents on an observer's ability to extract key awareness information from a filtered video sequences These filters were chosen because the can be performed within real-time constraints.

Observers will view two video scenes; each scene will be filtered using either the pixelizing or the smoothing filter. Scenes will use zero or more actors. Actors will be engaged in activities such as: eating; talking on the telephone; working alone; or, engaged in conversation. Typical work locales will be featured, such as private offices, coffee rooms, corridors, laboratories, conference rooms, and home offices.

The testing protocol is as follows. The observer views the video sequence with the filter applied across its entire range of extents, starting at maximum distortion. He answers a set of open and closed questions about the location, the actors in it (if any), their activities, and their state (e.g., disposition). For each question, the observer is asked to mention how it is he came up with his answer (i.e., from which features in the image sequence does he infer his answer), and how confident he is that his answer is correct. When the questions for one filtration level are complete, the level of filtration is reduced and questioning repeats. The process continues until the video becomes undistorted.

To elicit qualitative information about the observed filter's capacity to preserve one's privacy, at the conclusion of the tests for one of the filters, we ask the volunteer to imagine himself as the actor in the scene viewed, performing the same activities. When then ask him to choose the level of filtration at which he feels comfortable letting certain classes of people (e.g., strangers, close friends, superiors) view. The conclusion of an entire test session will be marked by an informal interview with open-ended questions about filtration and privacy.

As the study is just underway, results are few. Results from pilot tests clearly show that a single extent of filtration is insufficient to cover all privacy-awareness concerns: filtration must adapt to the demands of circumstance. Ostensibly, this makes sense: in face-to-face contact, the resolution at which we see another person and the understanding of their availability we derive from looking at him improves as we near him.

Additionally, it has become clear in pilot testing that motion in video is a critical source of information about presence and activity. Observers have great capacity to infer availability from subjective assessment of the motion they see. Effective filters should not destroy the continuity of motion throughout a video sequence, particularly if the underlying video is at a low frame rate, where changes resulting from rapid motion can be dramatic.

Pilot-test volunteers responded that they felt that the filters tested would be effective at preserving privacy, but many still wonder why one wouldn’t turn the system off entirely in such sensitive situations.

7. NEXT STEPS

Contingent upon the results of the present study, we hope to incorporate a privacy-preserving distortion filter into a teleawareness support tool. We next intend to examine lightweight techniques for controlling the level of filtration, such as reacting to information accrued from physical proximity detectors, motion recognition and other sources of teleawareness. Other issues to be examined include how to handle privatization/republications of sub-conversations in multiparty video media spaces, and how to smoothly integrate with other communication media and with workspace tools.

At present, we have made a simple networked, two-party video media space application that incorporates the pixelize filter at three different qualities of service: high frame rate, no distortion; high frame right, moderate distortion; low frame, high distortion. Figure 4 shows the highest and lowest qualities of service offered.

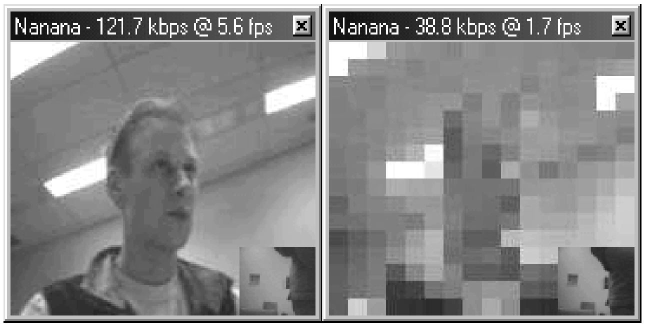


Figure 4. Highest and lowest qualities of service offered in our video media space app. Utilizes the pixelize filter.

This program demonstrates many important principles learned from previous research. First, it is reciprocal: the quality of service viewed by one party is the same as that viewed by the other. Second, quality of service is implicitly controlled. We suppose that placing the mouse pointer inside the video window is an implicit cue that one is engaged in interactivity with the remote party, and so higher qualities of service should be used. We do not know if this is an appropriate technique: although possibly interesting, it is largely intended as a placeholder for some other implicit control mechanism.

Third, changes in quality of service are mutually decided. One party may raise or lower the quality of service one level, but both parties must cooperate to take it to either extreme. Table 1 describes the behaviour of the quality of service control featured.

Pointer positions	Frame rate	Filter level
Both inside	High	No distortion
Inside and outside	High	Moderate distortion
Both outside	Low	High distortion

Table 1. Mapping of pointer positions (relative to each party's video window) to quality of service factors.

To discourage turning the system off to assure total privacy, the system allows one to "block" his video. While one party blocks his video, he sees the image of the back of a hand superimposed over the remote party's video image; the remote party instead sees the image of the palm of a hand. Both parties may block at any time—indeed even at the same time—but each is responsible for removing the block that he raises. We have attempted to give blocking a tangible aspect: it is toggled on or off by using one's hand to cover the camera so that one's image goes black for a few frames. We have chosen this approach to closely resemble one's tendency to reach for the camera in "dire" circumstances. Figure 5 shows the blocked video, as seen by the blocking and blocked parties.

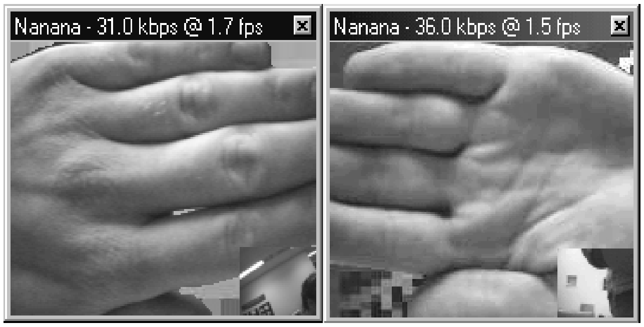


Figure 5. Blocked video as seen by blocking (left) and blocked (right) parties. Note video in corners can still pass through.

This video media space application is still quite experimental. The decision to use the pixelize filter was made arbitrarily, and will be revisited when the present study completes. The implicit and explicit interaction techniques have not undergone user testing. Many important technical problems relating to the capture, compression, and transmission of live video have been resolved and we are presently able to rapidly develop video-based applications using an iterative approach.

Lastly, more work is also needed in the field of video-mediated telepresence and teleawareness. It is not yet known conclusively if video provides any useful teleawareness or telepresence cues over and above other sources (e.g., on-line presence). Moreover, the minimum quality of service (e.g., frame size, rate, latency) needed to provide telepresence, and the maximum useful quality of service are still not known. In our present course of research, we hope to turn our attention to some of these issues.

8. ACKNOWLEDGEMENTS

Funding for this research provided by the National Science and Engineering Research Council, and Microsoft Research.

9. REFERENCES

[1] W. Buxton. Living in augmented reality: Ubiquitous media and reactive environments. In *Video Mediated Communication*, Hillsdale, NJ, 1997.

[2] A. Cockburn and S. Greenberg. Making contact: Getting the group communicating with groupware. In *Proceedings of the ACM/SIGOIS Conference on Organization Computing Systems (COOCS'93)*, pages 31–40, California, 1993.

[3] P. Dourish and S. Bly. Portholes: Supporting awareness in a distributed work group. In *Proceedings of the ACM/SIGCHI Conference on Human Factors in Computing Systems (CHI'92)*, pages 541–547, Monteray, CA, May 1992.

[4] R. S. Fish, R. E. Kraut, R. E. Rice, and R. W. Root. Evaluating video as a technology for informal communication. In *Proceedings of the ACM/SIGCHI Conference on Human Factors in Computing Systems (CHI'92)*, pages 27–34, Monteray, CA, May 1992.

[5] W. Gaver, T. Moran, A. MacLean, L. Lövstrand, P. Dourish, K. Carter, and W. Buxton. Realizing a video environment: EuroPARC's RAVE system. In *Proceedings of the ACM/SIGCHI Conference on Human Factors in Computing Systems (CHI'92)*, pages 27-34, Monteray, CA, May 1992.

[6] S. Greenberg. Peepholes: Low cost awareness of one's community. In *Proceedings of the ACM/SIGCHI Conference on Human Factors in Computing Systems (CHI'96)*, pages 206–207, Vancouver, BC, April 1996.

[7] C. Gutwin. Workspace awareness in real-time distributed groupware. PhD thesis, Department of Computer Science, University of Calgary, pages 11–31, 1998.

[8] ICQ Inc., *ICQ Version 99b Version b.3.19*, ICQ Inc., 1999.

[9] R. Kraut C. Edigo, and J. Galegher. Patterns of contact and communication in scientific research collaboration. In *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW'88)*, pages 1–12, New York, NY, 1998.

[10] Microsoft Corp. *MSN Messenger*, Microsoft Corp., 1999.

[11] R. W. Root. Design of a multimedia vehicle for social browsing. In *Proceedings of the Conference on*

Computer Supported Cooperative Work (CSCW'88), pages 25–38, New York, NY, 1988.

[12] I. Smith and S. E. Hudson. Low disturbance audio for awareness and privacy in media space applications. Technical report, Graphics, Visualization, and Usability Center, Georgia Institute of Technology, Atlanta, GA, 1995.

[13] J. C. Tang, E. A. Isaacs, and M. Rua. Supporting distributed groups with a Montage of lightweight interactions. In *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW'94)*, pages 23–34, Chapel Hill, NC, October 1994.

[14] J. C. Tang and M. Rua. Montage: Providing teleproximity for distributed groups. In *Proceedings of the ACM/SIGCHI Conference on Human Factors in Computing Systems (CHI'94)*, pages 37–43, Boston, MA, April 1994.

[15] Q. A. Zhao and J. T. Stasko. Evaluating Image Filtering Based Techniques in Media Space Applications. *In Proceedings of the Conference on Computer Supported Cooperative Work (CSCW'98)*, pages 11–18, Seattle, WA, 1998.

Designing an Integrated Bookmark / History System for Web Browsing

Shaun Kaasten
Department of Computer Science
University of Calgary
Calgary, Alberta, CANADA T2N 1N4
Tel: +1-403-220-7686
kaasten@cpsc.ucalgary.ca

Saul Greenberg
Department of Computer Science
University of Calgary
Calgary, Alberta, CANADA T2N 1N4
Tel: +1-403-220-6087
saul@cpsc.ucalgary.ca

ABSTRACT

Current commercial web browsers such as Netscape Navigator and Microsoft Internet Explorer attempt to make it easier for users to return to previously visited web pages. They offer three separate but important facilities: the back button, a bookmark system, and a history list. However, research indicates that users are not utilizing all of these systems effectively. In this paper, we present a single integrated history that unifies functionality similar to the back button, bookmarks and history lists. We also show how these mechanisms can be improved upon. We pay special attention to the ways pages are presented within a display, and on providing lightweight means for marking, recalling, and revisiting pages. The prototype we are developing works within Microsoft Internet Explorer. It presents a history list that represents pages by visual thumbnails as well as titles and URLs. It incorporates bookmark functions through a dog-ear metaphor, and it contains methods to filter a history list. Finally, it is tightly linked with the back/forward buttons.

Keywords

History, page revisitation, reuse, navigation, browser design, world wide web, hypertext.

1. INTRODUCTION

As the Internet continues to grow, the World Wide Web is becoming a mainstream information medium. People are relying on it more than ever for education, business and entertainment purposes. However, the World Wide Web is both enormous and largely unorganized, making it difficult to navigate effectively.

We all know how difficult it can be to find specific information on the web. One solution is better Internet Search Engines. This has become so important that the major players on the Web are in a race to provide effective search engines that can find what people ask of them, where the winner will get the most people and the most advertisers.

From the user's perspective, there is another problem that is getting less attention: how to return to information they have previously found. While a search engine might find the desired needle in the web's haystack, finding the same needle in the haystack later often requires them to search or navigate again, often with laborious efforts. Relocating a web page viewed days, weeks or even hours ago can be a frustrating and fruitless experience. Of course, current commercial web browsers such as Netscape Navigator and Microsoft Internet Explorer are not blind to people's need to return to previously visited pages. Each provides almost identical functions for returning to these pages, with the major ones being `Back`, the history list, and bookmarks.

In theory, these mechanisms should be heavily used, for almost 60% of all pages a person visits are to ones that they had seen previously [9]. Yet research indicates several problems with these mechanisms. While `Back` is heavily used, people have an incorrect model of how it works, which leads to surprises when just-visited pages are no longer reachable through it [6, 7]. Bookmark and history systems are used infrequently in practice [1, 9]. We believe that one of the reasons for these problems is that browsers provide revisitation systems in a fragmented, un-integrated, and heavyweight manner. `Back,` history and bookmarks all use dissimilar underlying models, different interfaces, and various ways of sorting and presenting groups of candidate pages. Recognizing pages in history lists can be problematic. Remembering to bookmark a page and managing the clutter of pages in a bookmark list can be onerous.

In this research, our goal is to integrate the idea of `Back`, history and bookmarks into a single integrated revisitation system that captures the best features of existing systems while remedying their known deficiencies.

2. CURRENT SYSTEMS

2.1 Bookmarks

Bookmarks have been an important feature, included in even very early web browsers. The concept is simple – when a user finds a page worth returning to, he or she asks the web browser to create a bookmark for it. Some time later, the user selects the bookmark to get back to the particular page. Unfortunately, the system fails in practice. While users are consistently return to previously visited pages, they rarely do this through bookmarks. The Tauscher and Greenberg study found that bookmark navigations only made up 2% of all navigation [9]. This can be traced to some major shortcomings of bookmark systems.

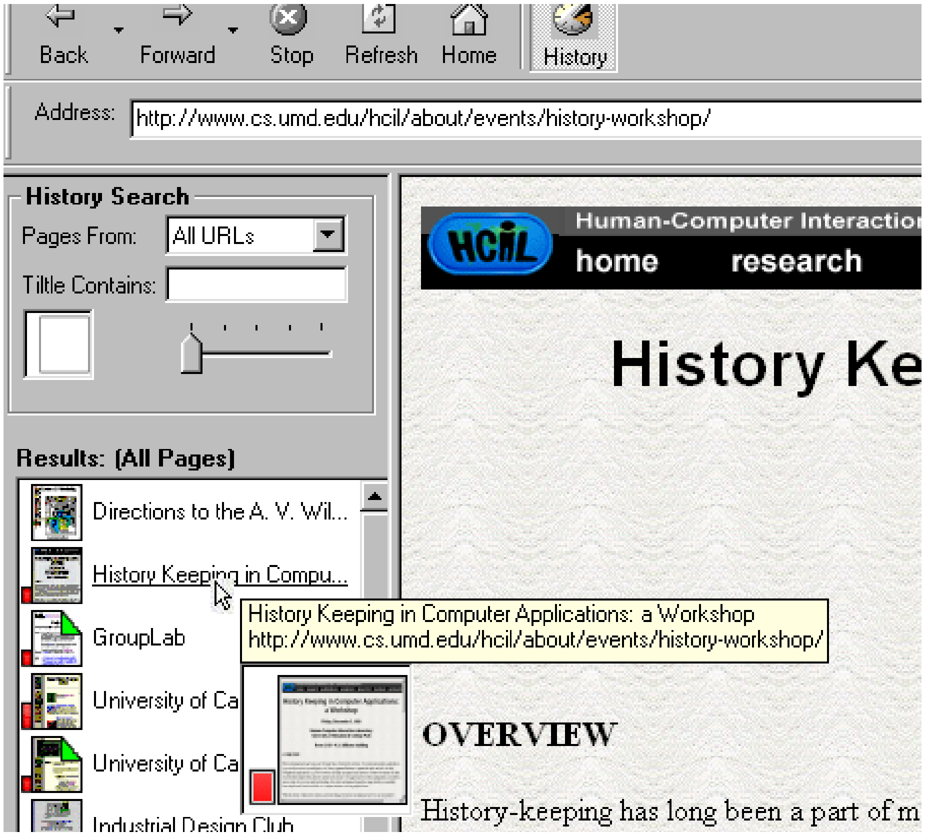
First, bookmarks are inefficient. A study on bookmark usage found that users continually accumulate bookmarks, which remain long after they are actually needed [1]. Thus, the bookmark collection becomes cluttered with unimportant items that inhibit the user from seeing the newer, now important bookmarks. In an attempt to reduce this problem, bookmark systems allow the creation of folders for organization. However, the study found that most users put off organization until it becomes absolutely necessary. Quite Simply, people are unwilling to devote the time needed to keep bookmarks organized.

Second, bookmarks are based on the notion that people know which pages they will need to return to. This requires a person to immediately determine the value of a page. However, circumstances can change, and a page that one decided to skip through last week can suddenly be valuable today. Users are poor predictors of what will be important at a later time.

2.2 History Lists

Web browser designers have recognized that people need a way to return to previously visited sites that have not been explicitly bookmarked. All contemporary browsers offer a fairly similar solution – the history list. A history list presents the user with a sorted list of every web page visited. The list can be sorted by the page title, URL address, the date of last visit, and even by number of times visited.

Like bookmarks however, history lists break down in real-world use. The Tauscher and Greenberg study found that users do not take advantage of the history system, as it accounted for only 1% of web page navigation [9]. Like bookmarks, this finding is related to some shortcomings of current history systems.



Address: http://www.cs.umd.edu/hcil/about/events/history-workshop/

Pages From: All URLs

Tiltle Contains:

History Keeping in Computer Applications: a Workshop
http://www.cs.umd.edu/hcil/about/events/history-workshop/

Figure 1 – Our prototype history revisitation system

The history list in early browsers had a great weakness- it was largely hidden from the user. Raising the history list required the user to select an option from a rarely used menu. While some users were not even aware of the option's existence, others found the process of uncovering it from the menu very inconvenient. As well, the history list was in a separate window that incurred the added work of window management. Current browsers have attempted to bring the history list closer to the user. Microsoft's Internet Explorer 5, for example, provides a large history button along side the more familiar controls, encouraging the curious user to try it out (see top-right of Figure 1). It also includes it as a tiled window, called an 'explorer bar' (similar to the left panel in Figure 1), which lives within the main browsing window.

When a user does uncover the system, he or she must then try to find the desired page within the list. This could be problematic.

To begin, web pages are often represented by title. Studies have pointed out how web page titles are frequently meaningless to the user [4]. First, it is the web page author, not the user who creates it. The author-supplied title may not match how a user thinks of that page. Second the web page author often gets it wrong: they may forget to include a title, or they may give a page the same title they had given to other pages within a site. This could be by accident (e.g., as when a titled page template is used over and over again and the author forgets to change it), or by intent (e.g., a title like 'FAQ' may identify a site full of 'FAQ's). When the user is confronted with a list of such unreliable titles, it can be virtually impossible to recognize the needed page.

Web pages can also be identified by their URL address. This address allows an individual web page to be instantly accessed from anywhere in the world. While these addresses are globally unique, they are often long and obscure. Users cannot be expected to memorize most URLs, and even recognizing a page from its URL can be difficult. In the best cases, the URL offers clues about a web page, especially when it contains a recognizable name such as Microsoft. Even here, the small display space given to the history list may mean that the entire URL is not visible to the user. In the worst case, the URL is meaningless (such as those produced automatically by databases). Thus, a history list of URLs is, at best, a hit and miss proposition for most users [4].

3. PROTOTYPE SYSTEM

We are currently developing a prototype system that works within Microsoft Internet Explorer. Some of our ideas were first explored in earlier versions [4, 5]. Figure 1 illustrates one of our prototypes. The unifying principle of this prototype is that all visited pages are organized as a recency-ordered history list, with duplicate pages shown only in their latest position.

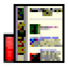
3.1 Back Button Integration

Our first challenge was to integrate the back and forward buttons with the history list. All commercial browsers implement a stack rather than a recency ordered list. We altered the back and forward buttons so that they work on a recency-based model instead of a stack: the buttons are thus just shortcuts for moving up and down the history list [4]. By bolding the currently visited item on the list, users see immediate navigational feedback on their `Back` and `Forward` actions. Aside from the fact that `Back` and `Forward` are now integrated with history, there are several advantages [7]. First, a person can always return to a page with a recency-ordered back button because, unlike stack, pages are not pruned from the list. Second, people who were asked how the conventional stack-based back button works all had the incorrect model that it was based on a recency ordered list [6]. Thus our new back button reflects people's model of how they *think* `Back` should work.

3.2 Implicit / Explicit Bookmarks

Our next challenge was to integrate bookmarks into the history list. We have transformed bookmarks into two different features.

First, *implicit bookmarks* are pages that are visited frequently: these pages are automatically visually distinguished from pages visited rarely. Both Figure 1 and the small inset here shows how we mark these pages with a vertical red band, where the height of the band indicates the degree of the page's visit frequency. If there is no band, the page has been visited only once or twice. Modest sized bands often indicate a 'hub' page i.e., a page that contains many children pages that have been visited. Since a user typically returns to the hub page to chase the next link [9], these typically show a higher number of hits. Large bands often indicate important pages, such as home pages for frequently visited sites.

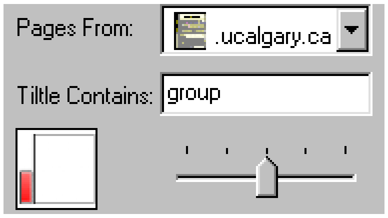


Second, *explicit bookmarks* are just pages on the history list that are tagged as special: we call these *dog-ears*. A user can quickly set a dog-ear by right-clicking over a page on the history list and selecting 'dog-ear' from a popup menu; users then have the opportunity to change the page's title if they wish. Dog-eared pages look different from non-dog-eared pages. For example, in Figure 1 (as well as this inset) we see that the 'Grouplab' and the second 'University of Calgary' page are dog-ears, as illustrated by the folded down corners in the upper right corners of the images.

3.3 Search Filters

To make this all work smoothly, users can perform dynamic queries [8] on the history list. By moving a slider (illustrated at the bottom of Figure 2), they rapidly and continuously filter the history list. As the slider is moved to the right, pages with low visit frequencies are immediately filtered from the list. When it is almost all to the right, only high frequency pages and dog-ears are visible. When positioned at the extreme right, only dog-eared pages are displayed: this transforms the general history list into a recency-ordered bookmark list.

We also have added a *search filter*, where the history list is filtered to display only those pages whose title contains the specified sub-string. As before, this is done through a dynamic query: as letters are typed, the list is immediately filtered to show only those matching pages.



Pages From: .ucalgary.ca

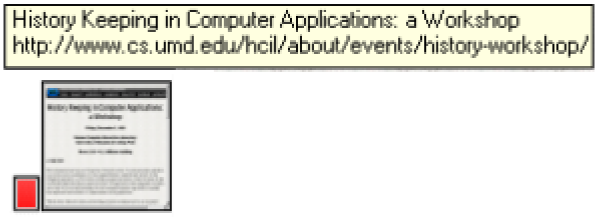
Tiltle Contains: group

Figure 2 – Search filters

3.4 Page recognition

Finally, we try to make scanning the list for a particular page easier by representing pages not only by title and URL, but by a thumbnail image as well [4]. Whenever a browser completely loads its page, we automatically do a window capture on it. We then scale the image into a modest sized thumbnail and store it on the local file system. Within the history list, we show a small version of this thumbnail (as visible in the Figure 3).

Additional detail is provided as a tool tip. As a person moves over a particular item in the list, we immediately display the page's full title, URL, and full-sized view of the thumbnail (see Figure 3).



History Keeping in Computer Applications: a Workshop
http://www.cs.umd.edu/hcil/about/events/history-workshop/

Figure 3 – Full-sized thumbnail and tool tip

4. USER TESTING

We are currently integrating this prototype into the standard Internet Explorer web browser. When this is complete, we will perform several studies on it. First, we will see how users perceive the differences between the standard stack-based back/forward buttons and our new recency-based model. Second, we will investigate how well users match web pages to the title, URL and thumbnail representations. Finally, we will conduct a field study by distributing our prototype system to volunteer users for use on their regular web needs. The browser will be instrumented to record the user's actions, just as previous studies have done [1, 9]. The analysis of these recordings will suggest which facets are utilized and well suited to the users' needs.

5. CONCLUSIONS

Studies have found that revisiting web pages is a frequent and important action. However, these studies also found that users of today's web browsers underutilize the functions to support this action. Our research suggests that today's web browsers can be improved by integrating these functions into a single system. We are developing a prototype incorporating this idea, and will soon begin our user evaluation process.

6. ACKNOWLEDGMENTS

NSERC Canada and Microsoft Research provided partial funding for this work.

REFERENCES

[1] Abrams, D., Baecker, R. and Chignell, M. Information Archiving with Bookmarks: Personal Web Space Construction and Organization. In Proceedings of the ACM/SIGCHI Conference on Human Factors in Computing Systems (CHI’98), pages 18-23, 1998.

[2] Ayers, E. and Stasko, J., Using Graphical History in Browsing the World Wide Web. In Proceedings of the Fourth International World Wide Web Conference, Dec 1995.

[3] Catledge, L and Pitkow, J., Characterizing Browsing Strategies in the World Wide Web. In Proceedings of the Third International World Wide Web Conference, April 1995.

[4] Cockburn, A., and Greenberg, S. Issues of Page Representation and Organisation in Web Browser’s Revisitation Tools. In Proceedings of the Australian Conference on Human Computer Interaction (OZCHI'99), November 1999.

[5] Cockburn, A., Greenberg, S., McKenzie, B., Smith, M. and Kaasten, S. WebView: A Graphical Aid for Revisiting Web Pages. In *Proceedings of the Australian Conference on Human Computer Interaction (OZCHI’99)*, November 1999.

[6] Cockburn, A. and Jones, S. Which way now? Analysing and easing inadequacies in WWW navigation. In *International Journal of Human-Computer Studies* 45(1), 105–129, 1996.

[7] Greenberg, S. & Cockburn, A., Getting Back to Back: Alternate Behaviors for a Web Browser’s Back Button. In *Proceedings of the Fifth Conference on Human Factors and the Web*, June 1999.

[8] Shneiderman, B., Williamson, C. and Ahlberg, C. Dynamic Queries: Database Searching by Direct Manipulation. In *Proceedings of Conference on Human Factors in Computing Systems (CHI’92)*, 1992.

[9] Tauscher, L. and Greenberg, S., How People Revisit Web Pages: Empirical Findings and Implications for the Design of History Systems. In *International Journal of Human-Computer Studies*, pages 47(1), 97-138, 1997.

[10] Hill, W.C. and Holland, J.D., Edit Wear and Read Wear. In *Proceedings of the ACM/SIGCHI Conference on Human Factors in Computing Systems (CHI’92)*, 1992.

Awareness Projected:
Moving Awareness to a Public Space

Michael Rounding, Saul Greenberg and Sheelagh Carpendale
Department of Computer Science
University of Calgary
Calgary, Alberta, CANADA T2N 1N4
+1 403 220 6015
[rounding,saul,sheelagh]@cpsc.ucalgary.ca

ABSTRACT

This paper concerns the design of public displays that project casual awareness information, where passerby's quickly acquire a sense of who in a small group is around and available. After describing what we mean by casual awareness, we present several prototype designs. In particular, we take awareness information garnered from a small group inhabiting a physical environment and abstract it onto a scene. We use the Three-Dimensional Pliable Surfaces visualization technique to emphasize people's particular activities.

Keywords: casual interaction, awareness, visualization, distortion viewing, public displays

1. INTRODUCTION

Casual interactions are the spontaneous and one-person initiated meetings and encounters that occur over the course of the day [10]. It happens so easily and naturally that it is something that most of us take for granted.

For casual interaction to work, people must be aware of what is going on in their environment: who is around, what others are doing, and seeing how available people are for conversation. People often gather this *casual awareness* information unconsciously or at the periphery of attention. For example, we hear the sound of an office door opening, which may indicate that someone down the hall has just arrived. Approaching footsteps indicate that someone is coming, and a quick glance (or our peripheral vision) will note whom it is. When we walk down a hallway we acquire a subtle but detailed record of "who is in" or "who is available."

Traditional casual awareness can only work when people inhabit a common space. Kraut, Egido and Galegher [10] correlated the number of interactions between people as a function of their proximity to each other: their results dramatically illustrate that the incidence of casual interaction drops exponentially with distance. People sharing offices have high amounts of casual interaction; people whose offices are separated by tens of feet have lesser interactions; and people in different floors of the same building have far less encounters.

While conventional workplace design often places potential collaborators in a common space (thus increasing casual interaction), this is not necessarily the case in new workplaces. Large distances may separate collaborators: across buildings, cities, or continents. Rapidly changing work demands may also require quickly changing relationships: a new co-worker required for a current project may be located on a different floor. Even if it makes sense for them to share a space, the cost and constraints may make this impractical.

Consequently, there has been much work in supporting causal interaction between pairs of distant collaborators, usually through media spaces or other information displayed on a personal computer screen e.g., [1]. In contrast, our research goal is to support casual interaction between small groups of people by projecting casual awareness information into a public space via a computer–controlled medium. We expect to present types of casual awareness information similar to what now exists in a physical work environment, where people can gather and maintain this information at the periphery of their attention.

We begin in Section 2 by exploring related concepts of awareness, including situation awareness and workspace awareness. In Section 3, we introduce a real scenario where casual interaction in our work-a-day world was breaking down: we will use this scenario to motivate our investigations. Section 4 then takes a detour into an information visualization technique called three-dimensional pliable surfaces. In subsequent sections, we show how this technique has potential to be used as a method to unobtrusively display casual awareness of a group.

2. AWARENESS

Awareness can be most simply defined as 'knowing what is going on in an environment' [6], which 'involves states of knowledge as well as dynamic processes of perception and action' [9].

More specifically, Gutwin and Greenberg [9] identify four basic characteristics of awareness.

1. Awareness is knowledge about the state of some environment, a setting bounded in time and space.
2. Awareness knowledge must be kept up-to-date: because environments change over time, a person must continually gather and update what they already know.
3. Awareness is gathered from the environment both through sensory perception and by actively acting upon the information acquired.
4. Awareness is rarely the primary goal of a person; rather, it is almost always part of some other activity.

There is a history of research in awareness, although particular researchers have tended to focus on specific situations and thus different awareness demands. Because much of it applies to casual awareness as well, it is worth reviewing them. Two particular sub-areas are visited here: situation awareness and workspace awareness.

Situation awareness can be defined as the up-to-the minute cognizance required to operate or maintain a system [6, 9]. The maintenance of this information can be broken down into three steps:

- being able to discern information in the environment that is relevant to the specific task being performed
- taking the perceived information and integrating it correctly into the previously existing knowledge of the system
- being able to predict how this information could change in the near future

Situation awareness has been explored heavily by human factors specialists who look at how highly trained machine operators act on awareness cues in an information rich and often stressful environment: jet fighter pilots, surgeons, and so on.

Gutwin and Greenberg [9] re-defined the work in situation awareness into a more specialized instantiation of awareness called *workspace awareness*, where several individuals require awareness to efficiently operate and interact with each other when using a shared visual workspace (such as tabletops, whiteboards and groupware drawing surfaces and editors). Workspace awareness is formally defined as 'the up-to-the-moment understanding of another person's interaction with the shared workspace' [9].

Gutwin and Greenberg [9] go on to describe how people maintain workspace awareness by using a modified version of Neisser's [11] *perception-action cycle* of how information in a physical setting is gathered and interpreted. Quoting from Gutwin and Greenberg: 'people gather perceptual information from the environment, integrate it with what they already know, and use it to look for more information in the workspace'. People gather information through three means: *consequential communication* (the presence of hands and bodies in the workspace); *feedthrough* (movement and changes to workspace artifacts), and from *intentional communication* between participants (both verbal and gestural).

These notions of situation and workspace awareness suggest a more formal way of viewing casual awareness: while we expect the perception/action cycle to remain the same, the sources of information will likely differ somewhat. This is still work we have to do, but we are particularly interested in how new arrivals in an environment acquire knowledge of what is going on, that is, how they get an entry point to the perception / action cycle.

To explain this further, we will describe in the next section a breakdown in an actual casual awareness situation. We will describe how the original breakdown occurred because of the lack of an entry point into awareness, and then how adding an entry point partly but perhaps inadequately repaired it. This will be used as a context to suggest an interesting and useful research direction.

3. SCENARIO: A BREAKDOWN IN CASUAL AWARENESS

We present a problem that we observed in our own physical workplace, a problem that we believe occurred because there was no 'entry point' to casual awareness information. We use this problem to describe a possible solution, and to motivate our research.

3.1 The Scenario and Its Problem

GroupLab members, mostly graduate students, occupy a shared room full of dedicated workstations (the GroupLab laboratory). The lab is set up in such a way that a person looking in from the doorway may see some but not necessarily all of the people who are present. The layout of the workstations could block their view, and there are two additional rooms attached to the lab that cannot be seen from the doorway.

Throughout the course of a fall semester, all of the members of the lab were involved in many different activities. These activities saw them coming in and out of the lab at unpredictable intervals. Some of the activities occupying the students' time included teaching assistant positions in undergraduate level courses. With these positions came frequent visits by undergraduate students seeking consultation outside of class time.

Undergraduate students would arrive in the lab looking for their teaching assistant. They would typically glance in and find them not within their immediate line of sight. For this reason, they sought the assistance of other members of the lab in locating the desired person. The members of the lab who had been interrupted would then inform the student that their teaching assistant was either present, or that they were not currently in the lab. In the latter case, they would often make an educated guess as to when or if the person sought after would return.

The problem here should be fairly obvious. Information that was immediately apparent to those already occupying the lab space was not apparent to new arrivals in the environment. The new arrivals had to poll occupants of the system for information to gain a fast entry point into the perception cycle. This polling interrupted the workflow of the lab occupants temporarily: they were forced away from their working line of thinking in order to summarize their current state of casual awareness of the desired GroupLab occupant.

As often happens, an ad-hoc solution was found. Near the end of the semester, the lab adopted a somewhat public notification of who was around. As pictured in Figure 1, below, it was basically a picture of each member of the lab, accompanied by textual descriptions of several places that they could possibly be or things that they could be doing. Using a post-it note stuck to the one of the descriptions, each person could communicate their state (i.e., in or out) and what they were doing (i.e. where they were) to anyone who was looking for them at any particular point in time. Of course, there is nothing new about this approach: we often see individual offices where its occupants place similar notices on their door.

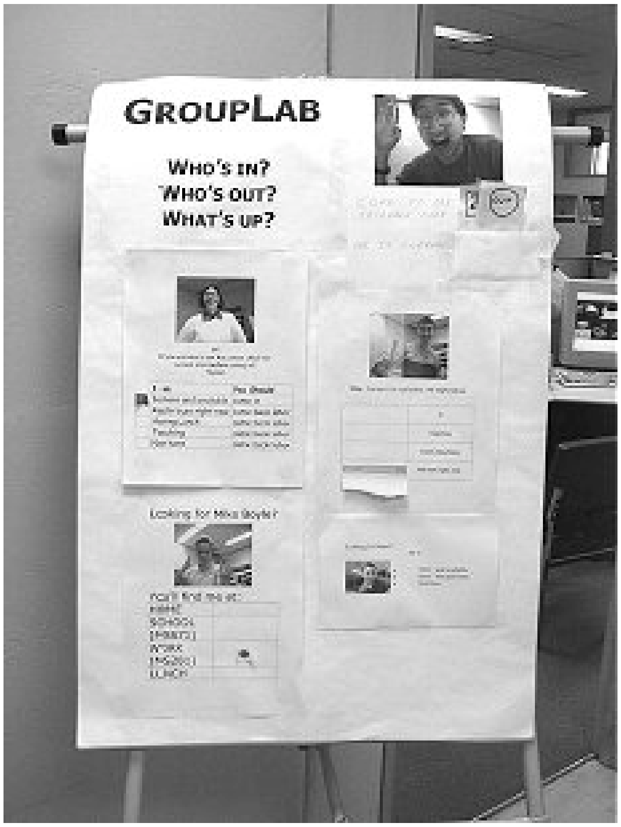


Figure 1: our laboratory's simple solution to the breakdown in casual awareness

This system reduced the interrupting traffic in the lab somewhat but, unfortunately, was still not very noticeable to visitors. It also required lab members to constantly maintain the information panel. Forgetting to do so meant that visitors had the wrong information (e.g., the person is shown as absent when they were actually there). This eventually led to mistrust i.e., even though the information panel would say one thing, visitors would still look inside to confirm it.

3.2 A Possible Solution

This example introduces several design challenges if we are to automate a system for casual awareness. First, we need to track and capture automatically casual awareness information about people: if the system requires explicit acts (such as pressing an 'I am here' button) it will likely fail. Second, we need to present the information about the state of all occupants in the lab in an effective manner: if visitors cannot immediately interpret the display to know what is going on, they will bypass it and go into the lab instead. Third, we need to display information captured over time, where visitors could immediately discern not only the state of the system, but also (if necessary) how it got to that state. For example, a GroupLab occupant may know that one of its members is around (because they have been in and out frequently in the last hour) even though they may not be present at that moment.

Our approach is to build a public artifact—where information held by the computer can be seen, heard, and acted upon in an easy and natural way by the people inhabiting the area around the computer [7]—for casual awareness. We want the public artifact to provide a fast entry point to the perception / action cycle. This could be similar to the "who's around" board mentioned above, although we would have to design it not only to gather and present awareness information, but also to fit well into the physical environment. We also have to design it to be highly visible but still unobtrusive, where those occupying and visiting the space can be gather information at the periphery of their attention.

This description is derivative of the notion of *calm technology* [12]. *Calm technology* does not demand our attention as other things do. The information calm technology may present or represent is available, but unless it is consciously observed, it remains largely a peripheral to the rest of the environment it inhabits [7,12]. The device recedes into the background until some event attracts attention back to it. Located in a public space – such as the lab discussed in the previous subsection – such a device can become a public artifact that is, in effect, ignored until either needed or until something happens that attracts attention to it [7].

While this idea sounds simple enough, there are a number of issues to be resolved. For example, placement of public artifacts is extremely difficult: where to put something so that everyone can see or use it can vary greatly depending on environmental factors. We have to recognize that a device's physical location can encourage or discourage public interaction. Notes placed on a whiteboard are considered more public than notes attached to computer monitors. Similarly, a workstation monitor is a far more private artifact than is a large interactive computer whiteboard (in essence a projected computer monitor). Making information public and visible encourages interaction [7].

Making awareness information public and available can make it easier for new arrivals in an environment to immediately determine its state. The information presented in public view definitely needs to be useful. As our scenario illustrated, not just any and all information can be explored at once. In essence, three questions need to be answered:

1. What information does an entire group need to see?
2. When and for what is the information important?
3. How should that information be conveyed?

Answering these three questions is at the core of designing a casual awareness system. Of course, the idea of calm technology suggests a useful mechanism for tracking and presenting casual awareness information, while the idea of public artifacts forces us to think how such devices fit comfortably within a public space.

Although this describes the properties of the device that contains casual awareness information, we still do not know how to present that information as a display. The next section suggests one approach. It details briefly a visualization technique called three-dimensional pliable surfaces, which we will then apply to present casual awareness information.

3. 3-DIMENTIONAL PLIABLE SURFACES

One possibility for visualization is the use of distortion techniques that magnify some information at the expense of other information. The purpose of this section is to just give a simple overview of the visualization technique. We begin by describing fisheye views and how it tries to solve screen real estate problems. Three-dimensional pliable surfaces are then presented as an alternate solution to the fisheye approach. As we will see in Section 4, we will use this idea to make particular awareness cues more salient on a public display.

3.1 Fisheye views

The small screens and poor resolution of existing computer display often means that they are inadequate for displaying large amounts of detailed information in a single scene. This is the “screen real estate problem;” and the challenge is how to display more information onscreen in a useful manner [2].

When viewing a set of information, centering or zooming in on a particular point in the data eliminates the context that existed between the focus of the zoom (the point of interest) and the whole space. As an example, taking a map of a city and zooming in on a particular subdivision eliminates the sense of that subdivision’s location with respect to the rest of the city. This loss of context is the problem that distortion-oriented display techniques attempt to solve. One particular approach to maintaining focus plus context is *fisheye views.*

With fisheye views, all the data for a given data set is presented onscreen with a moveable focal point that magnifies the area underneath it in place using a fisheye lens effect (see Figure 2, below). With a large amount of detailed data present, displaying all of it at once in sufficient detail becomes difficult. Scaling to fit is inadequate. For example, text can quickly become too small to read. With a fisheye lens applied to a particular focal point, the area underneath the lens is magnified and thus visible in detail, while still showing its position and relation to other items in the global context.

Many fisheye implementations work best when the data being viewed is discrete, as they apply a fisheye effect only to the object under the focal point. When the data is one continuous entity (such as a map), a more sophisticated technique is required to achieve the same effect. One such technique is three-dimensional pliable surfaces, discussed next.

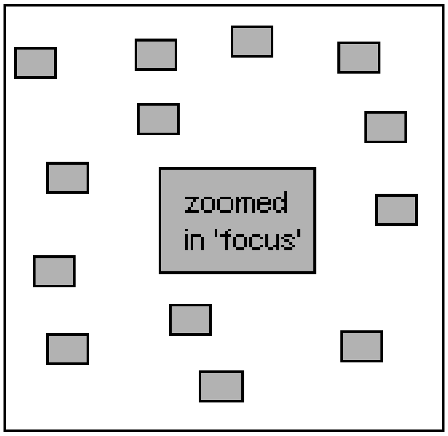


Figure 2: fisheye viewing

3.2 Pliable Surfaces

The three-dimensional pliable surfaces technique, developed by Carpendale et. al [2] applies a distortion effect to a two-dimensional surface by changing its shape from a plane to a three-dimensional surface. This visually distorts particular areas, which can be used to effectively draw attention to a particular region of some data representation while maintaining the context between the focus and its surroundings.

When applied effectively, the technique results in an increase in the amount of information that can be displayed usefully on a screen without getting lost. Accentuated distortions can easily provide a mechanism for drawing the eye to particular region. For example, we see a map in Figure 3A, and how the area in its centered has been distorted in the middle in 3B. The 3-d manipulation that stretches the surface effectively expands some regions and shrinks others. What is important is that the area under the lens is more visible. The grid lines in Figure 3B and the added shading in 3C are cues that help a person interpret what areas are distorted and what areas are undistorted.

To create the distortions, a two dimensional image is manipulated in three dimensions, pulling regions towards and pushing regions away from the viewer so as to accentuate certain elements within the image. This allows certain elements to be viewed smoothly in detail without actually losing their context within the whole of the information space.

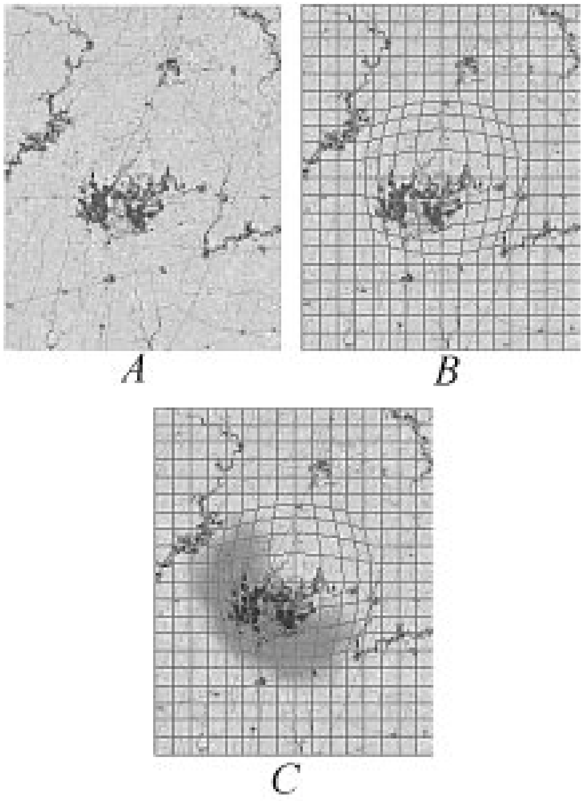


Figure 3: pliable surfaces: A: a simple distorted image. B: simple distorted image with an added grid. C: same image presented in B, with added shading.

A Gaussian curve is applied to a two-dimensional surface to create the distortion. The gently curving profile of the top and base of this curve makes it a good choice. When viewed in perspective, the fall-off of the Gaussian curve is compressed slightly. This is improved by using an auxiliary curve to adjust the

Guassian curve, making the middle of the drop-off much more gradual. Viewed from above, the distortion appears to be much less compressed than with just the Gaussian curve alone [2].

There are several advantages to this distortion approach that make it convenient and easy to use. For one, viewing such a transformed surface from above is akin to having a section of it pulled out of the screen, making the distorted region appear to be closer to the eye of the viewer. It is fairly natural for human beings to manipulate things in three dimensions because they are used to it. Pushing and pulling things makes sense in a real world; pushing and pulling regions in the described distorted world should also be easy to understand [2].

While some distorted images are easy to understand on their own, adding certain features to a distorted image can further imply where and what the distortion is. For example, adding a grid (e.g., Figure 3B) that contours the distortion outlines the bounds of the distortion, making it immediately more comprehensible. Furthering this idea, adding a light model to the model introduces shading and makes the distortion even more apparent to the eye (Figure 3C)[5].

In the next section, we outline the use of this distortion technique in a possible implementation of the casual awareness problem we have previously described.

4. AWARENESS PROJECTED

There are currently a few different ideas for presenting awareness information to others. Two popular approaches include ICQ programs and video media spaces.

Mirabilis’ ICQ®, AOL’s Instant Messenger®, the Microsoft® Instant Messenger®, and Greenberg’s Peepholes [8] show a list of a person’s preferred on-line contacts, their current login state, and how long it has been since they have touched their computer. Although this information is captured automatically by the system, people can explicitly change their status through simple control mechanisms. This adds a notion of control and privacy. For example, in Mirabilis ICQ a user can force their network status to “not available” or even to “invisible,” thus granting them a certain level of privacy. While these systems are extremely simple and use surprisingly crude mechanisms to capture information, they are very effective and popular. This is because its users immediately go from having no awareness of their colleagues to having some awareness, even though the underlying awareness information is unsophisticated.

Another way to present awareness information is through the use of video media spaces, an always-on audio and video channel between two or more locations [1, 7]. Because colleagues can see each other through this channel, they can fairly accurately track their availability. What is especially appealing about these systems is that the same channel is used for communication. That is, people can act on awareness simply by entering into conversation with the person on the other side. Of course, media spaces such as these have privacy issues associated with them.

While both of these types of systems are successful in conveying awareness information, neither has been developed as a public artifact. That is, they are mostly designed to be used by one person at a time. In media spaces, for example, the pure video information could be too rich to be displayed publicly, as privacy concerns can become very serious indeed [7]. The next two subsections suggest a couple of systems that could be interesting to have displayed publicly. Both use pliable surfaces, as described in Section 3, as a visualization tool to represent awareness.

4.1 Small Group Awareness

The level of awareness provided by ICQ-like systems simply present people’s state as a single data point, one per person. A given contact, listed by name or by image, is labeled as “connected,” “away,” “not available,” etc. The first problem is that seeing who is around means scanning individual items in the list, which means that maintaining awareness information would become a foreground rather than background activity. As well, the discrete data points declaring status are crude. For example, in the default configuration of Microsoft Messenger, a person is displayed as ‘present’ if they have touched their computer within a half-hour (or whichever time delay is specified by the person). This label is thus, at best, just an estimate of presence, and there is much room for error. The problem is that we need better ways to represent people’s presence.

We can use the pliable surfaces technique to present the information in a scene that can be interpreted at a glance, and to present the notion of presence as a continuous variable [7]. Our suggestion is to use a picture or series of pictures representing a group, where we magnify particular people by an extent that reflects their level of activity. For example, Figure 4A shows the group where no one is on line. In figure 4B we see that Michael’s head is larger (front row right), and thus he appears somewhat more present than the others. This works for multiple people as well: in Figure 4C we now see that the person at the middle row far right is also present. The degree of presence can be marked by the amount of magnification: people on-line now would be magnified more than people who have not done anything for a while. They would thus stand out in the scene.

In practice, we would build the actual application so that a group could import an image into it, and specify areas of the image to represent different people. When running, the distortion lens could be applied to the surface of the image based on some measure of activity.

As a public display, we could show this image on a large display situated in a reasonable place. In our lab scenario mentioned previously, this could be a 1m^2 flat panel display positioned on a wall or easel just outside our laboratory door for others to see. Unlike the current paper version, lab members would not have to update the display explicitly, and visitors or people just walking by could interpret the scene at a glance. While we recognize that some initial training / signage may be necessary to indicate to people what this new technology is doing, we believe that it will be easy to interpret afterwards.

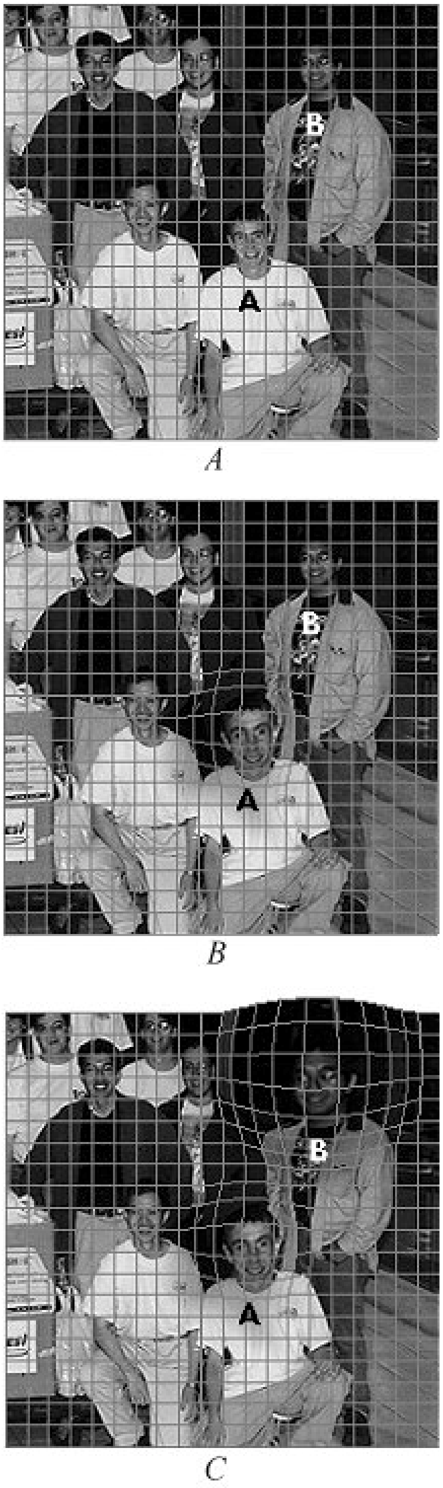


Figure 4: Distortion providing awareness. In frame B, person A has initiated some activity while in frame C person B has initiated some larger activity.

4.2 Awareness Maps of Physical Space

Another potentially interesting application of distortion techniques in an awareness application could be to use it to represent activity in a physical space on a two dimensional map. This sort of system could be implemented using motion sensors placed in locations relative to a floor plan.

For example, the representation illustrated in Figure 5 suggests some level of activity near the computer and a larger level of activity near the exit. Someone viewing this data would be able to determine that someone was probably using the computer and that someone was either entering or leaving the room. This later case would be clearly visible if the distortion effect were applied in real time, where the focal point would track the person as he or she moved through the room.

If the same technique were used to represent a lab, the proximity of people to computers or phones could be measured visually, as well as their level of activity (to a certain extent). For example, if a person were on a telephone we would see the telephone magnified as well. This could allow a visitor to make a more accurate decision about how to best go about interrupting or contacting the individual in the represented space. Of course, there are “Big Brother” implications here that have to be addressed.

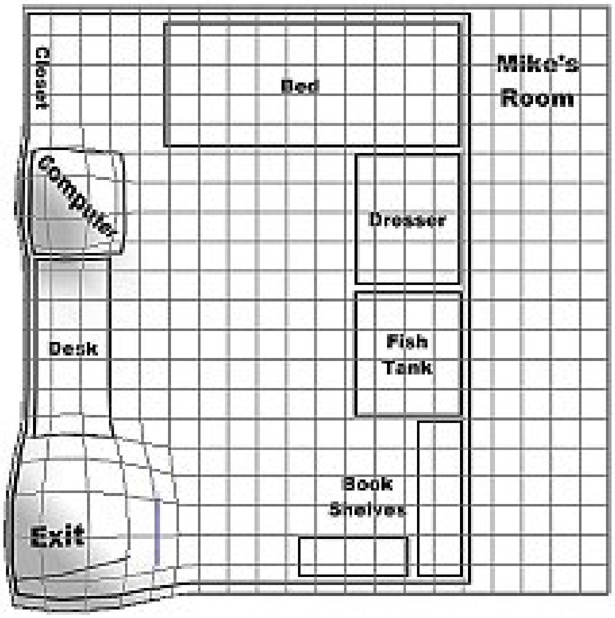


Figure 5: Activity in a physical space mapped and represented using a pliable surface.

5. CONCLUSION

This paper lays out a research direction. We summarized the casual awareness problem, and related how existing theoretic models of awareness can give us the intellectual foundations to understanding what is required of awareness systems. We then presented a particular problem in maintaining casual awareness, that is, how visitors and passersby’s can discover the state of people inhabiting a common area. Using this as our motivation, we introduced and then proposed a casual awareness display that uses the notion of distortion-oriented techniques and public displays.

The work is still in the early stages. The pliable surfaces system now exists [2], but is not in the form that it can be readily applied to this problem. We need to extend its API, and to craft an interface that lets users specify individuals in an image. We need to control the degree of magnification in real time, where sensors or other approaches can drive the degree of magnification. We need to explore the device characteristics itself, and understand architectural issues of how devices can be perceived as public artifacts. Of course, we need to evaluate all this. It is unlikely that this design is the best or even an effective approach. We need to understand where it succeeds, and where it fails. We expect new and better designs to come out of this as well as an understanding of the design issues associated with casual awareness displays.

6. REFERENCES

[1] Bly, S., Harrison, S. & Irwin, S. (1993) Media spaces: Bringing people together in a video, audio and computing environment. *Communications of the ACM*, 36(1):28-47.

[2] Carpendale, M. S. T., Cowperthwaite, D. J. and Fracchia F. D. (1995) Three-Dimensional Pliable Surfaces: For Effective Presentation of Visual Information. UIST'95, Proceedings of the ACM Symposium on User Interface Software and Technology, pp 217 - 226, ACM Press, 1995.

[3] Carpendale, M.S.T. (March 1999) A Framework for Elastic Presentation Space – Simon Fraser University

[4] Carpendale, M. S. T., Cowperthwaite, D. J. and Fracchia F. D. (1997) Extending Distortion Viewing Techniques from 2D to 3D Data. IEEE Computer Graphics and Applications, Special Issue on Information Visualization, IEEE Computer Society Press, Vol. 17(4) pages 42 - 51, July 1997.

[5] M. S. T. (1997) Carpendale, D. J. Cowperthwaite, and F. D. Fracchia. Making Distortions Comprehensible, IEEE Symposium on Visual Languages, pages 36 - 45, IEEE Computer Society Press, 1997.

[6] Endsley, M., (1995) Toward a Theory of Situation Awareness in Dynamic Systems, *Human Factors*, 37(1), 32-64.

[7] Greenberg, S (1999). Designing Computers as Public Artifacts. - *International Journal of Design Computing:* Special Issue on Design Computing on the Net (DCNet'99), November 30 – December 3, University of Sydney.

[8] Greenberg, S. (1996) Peepholes: Low Cost Awareness of One's Community, *ACM SIGCHI'96 Conference on Human Factors in Computing System, Companion Proceedings*, p206-207.

[9] Gutwin, C., and Greenberg, S. (1999) A Framework of Awareness for Small Groups in Shared-Workspace Groupware. - Technical Report 99-1, Department of Computer Science, University of Saskatchewan, Canada

[10] Kraut, R., Egido, C. & Galegher, J. (1988) Patterns of Contact and Communication in Scientific Collaboration – In Proceedings of the ACM Conference on Computer Supported Cooperative Work, 1988. pp 1-12.

[11] Neisser, U. (1976) *Cognition and Reality*, W.H. Freeman, San Fransisco.

[12] Weiser, M. and Seely Brown, J. (1996) Designing Calm Technology. *PowerGrid Journal,* Version 1.01, http://www8.electriciti.com/1.01/

Change Management

James Tam, Saul Greenberg, and Frank Maurer
Department of Computer Science
University of Calgary
Calgary, Alberta
phone: +1 403 220 3532

tamj@cpsc.ucalgary.ca

ABSTRACT

In this paper, we analyze the problems of keeping up with diagrammatic changes being made within a collaborative software design tool. With these graphical tools, one software engineer may specify possible software architectures e.g., through a UML editor. A second software engineer may then modify the diagram. The problem is how the original engineer can track what changes had been made. Most systems provide little or no support for this, and we believe that relying on the engineer's memory is inadequate. We propose a several graphical representations that can illustrate to an author what has changed. Issues in graphical representation of changes include how actual changes can be portrayed, as well as ways to filter the view to ensure the designers only see relevant changes at an appropriate level of detail.

Keywords: Change Management, information filtering, graphical representations, graphical software design tools.

1. INTRODUCTION

Teams involved in collaborative software design and engineering often follow a divide and conquer approach. They try to split the project into manageable tasks, where a single person can work on a task. The challenge is in bringing the deliverables together into a cohesive and testable whole.

However, there are also times when team members must work on the same task and task artifacts: documents, software, UML diagrams, functional specifications, and so on. One person may work on the artifact for a while, and then pass it on to the next person (perhaps to continue the work or to revise it as required). Iterative development can then occur between these two people in a back and forth manner, or it could even include other team members.

The problem is how team members can recognize and / or track changes made in a document by other members. At the crudest level, this could simply be a case where one person analyzes a changed artifact, perhaps relying on memory to recognize changes. Errors and inaccuracies are, of course, likely. Slightly better is the case where people communicate over the document, where one tells the other (perhaps orally or by notes within the document) what has been done. Of course, this requires a great deal of additional work, and it is easy for one person to neglect to tell another about some crucial change.

What is needed is a more formal way to support how people recognize changes within artifacts. We thus define *change management* as a process that helps one person recognize and track changes made by one or more others.

Our particular interest is in how change management can be applied within a diagramming tool, such as a UML editor. We are especially concerned with how changes within these tools can be tracked, how the relevancy of changes can be determined, and how changes can be displayed to another person in an effective and efficient manner.

We set the scene by first describing previous work on change management in both human computer interaction and in software engineering. We then discuss the difficulties of doing change management in a graphical diagramming tool, especially when many changes require some notion of filtering to reduce complexity. Finally we will present and discuss our early work in how to represent changes within a graphical editor.

2. PREVIOUS RESEARCH

2.1 Software Engineering research in change management

Within the field of Software Engineering, researchers have stated that the goal of change management is to be able to predict how a software project will be affected by the changes that are made to the project [1]. This is somewhat different from our view of change management: while we are concerned with *what* changes have been made, this other view considers the *effect* of changes.

Still there is work related to our own definition of change management. In particular, almost all programming environments contain some kind of version control system. One example is the CVS (Concurrent Version System) available in Unix, and Microsoft Visual SourceSafe available as part of the Microsoft Visual Development Suite. Both allow programmers to check in and check out versions of programs and other documents as they are being developed, and allow comments to be added to them (such as a textual note describing what has changed). They also allow differencing of versions, where differences in text sequences are shown.

Dellen [2] developed a different type of change management framework that would automatically notify the interested members of a team when particular changes occurred. It used an event-driven notification system: as a change occurred in a piece of software, an event would be raised. If developers within the project had registered their interest in that particular type of event, then they would be notified of the change. Similar to this, programmers hooked the Elvin notification server [3] into CVS, where notifications of how files were checked into and out of the repository appeared on a one line tickertape. As with Dellen's system, programmers could subscribe to those items they were interested in.

While these strategies are interesting, all are somewhat difficult to apply to the graphical nature of diagramming systems. Programmers may find it difficult to articulate a graphical change: in real life, we often gesture around the drawing to do this. This makes the current notification and comment annotation components of systems somewhat unwieldy. The version differencing tools only work on sequential text: they are not able to show differences within a graphical drawing, such as would be found in a graphical design editor for the Unified Modeling Language (UML).

Unlike these other systems, which are centered on sequential text, Rational Rose (by Rational Software Corporation) contains a change management facility that works in a UML diagramming tool. It works by translating the diagram into a hierarchical text description and by highlighting changed items within this text. Unfortunately, this representation of the UML diagram and its changes are no longer in graphical form: thus programmers must view it in a different (and perhaps more difficult to understand) representation. As well, an approach such as this would not be able to handle free form annotations and marks that can be added to the UML view, as can be done in the Argo open source UML editor [4].

2.2 HCI research in change management

As with software engineering, most of the previous research has focused on text based work environments. Perhaps the best example is Neuwirth et. al.'s "Flexible Diff-ing" text differencing system [5] which was developed by Neuwirth, Chandhok, Kaufer, Erion, Morris, Miller. What makes it special is that it allows viewers to contrast changes at various levels of detail. Thus changes can be viewed at a high level (e.g., where have changes been made) as well as in progressive detail (e.g., exactly what changes have been made).

Hill and Hollan [6] proposed one graphical approach related to change management called "edit-wear and read-wear". They would track what parts of a document had been either read or edited, and would then use graphical "wear" indicators to indicate how much had been changed and in what places. The more often that a portion of text was changed, the more vivid the wear indicator [6]. While there was a brief discussion of how some of these ideas might be applied to a graphical based environment, such as applying wear indicators to user interfaces, the main focus of the research was conducted in text systems.

3. CHANGE MANAGEMENT ON COLLABORATIVE GRAPHICAL DOCUMENTS

Our particular interest is how change management can be supported in a collaborative process that uses predominantly graphical rather than textual documents. While we are interested in how change can be tracked within all 2-dimensional graphical drawing and diagramming applications, we will concentrate for now on how software engineers collaboratively develop UML diagrams. Our approach is to somehow track and visualize changes within a diagram so that engineers can answer questions such as:

- Have any changes occurred since I last visited this document?
- How many changes have occurred?
- Where have these changes happened?
- How have particular parts of the diagram changed?
- Who did these changes?
- Why did they perform these changes?

These questions were derived from similar questions raised by Gutwin [7], who was studying how people would track what others were doing when working together in real time over a visual work surface. Gutwin was interested in what he called Workspace Awareness . While related, our own work will focus on awareness of changes in an asynchronous visual work surface designed for software development rather than a real-time type of system.

In the following section, we raise two issues that we believe must be addressed by any graphical change management system: information filtering and the techniques used to represent changes visually.

4. INFORMATION FILTERING

We expect some graphical documents to change little between versions, and some to change quite a bit. We expect cases where changes pervade the entire document, and others where they are quite localized. The problem is that in all these different cases, the viewer of the document must somehow make sense of what has changed. Showing all changes at all levels of detail may be confusing when many changes are present, and people will have to do much work in order to determine which of the many changes are relevant.

One solution is to apply information filtering techniques to change management. This involves having the system somehow screen all changes that have occurred, and showing only the important changes to designers. Particular changes may be shown at a higher level of abstraction i.e., the abstraction could indicate that an object has changed, without detailing all the changes within it. Without filtering, designers may become bombarded with volumes of changes that they may or may not always be interested in.

4.1 Filtering: Too much vs. not enough

An important issue that is immediately raised is determining which changes should be shown to designers and which ones should be hidden. If too many changes are filtered then there is a risk that important information may be lost. If too few changes are filtered then the person may be overloaded with irrelevant information.

There are two approaches for determining the relevancy of a change to a designer. The first way is to try to automate this process and having a program determine what is relevant for an individual. The second way is to allow people to decide the issue for themselves.

As previously mentioned, there was already some research conducted using the first approach by Dellen [2]. Her approach was tailored for situations where programmers were working on different but inter-related parts of a system, which differs from our situation where people are working on the same part of the system (or diagram).

In the second approach, the people would set their own criteria for determining relevancy. Only the changes that have meet these criteria would be displayed. For instance, a person may only be interested in seeing changes that occurred during a certain period of time, or seeing changes that were caused by a certain person. The problem, of course, is in giving people appropriate ways to determine relevancy within the tool.

5. GRAPHICAL REPRESENTATIONS

Assuming that a diagramming tool knew *what* changes are relevant to show to its user, it still must decide *how* to show these changes to the person. A key issue is the visual representation used. Any indicator of change must be noticeable enough so that it is easily interpreted and not overlooked, while remaining unobtrusive so that it does not interfere with the real work of software design.

The first step to finding good representations is to determine the classes of changes that can be made to a diagram. Because we are still in the early stages of our work, we have explored only three primitive change operations that people can apply to a UML diagram: the addition, deletion, and modification of objects.

There are, of course, many possible ways to represent these operations to a viewer. We are beginning our work with simple change indicators: icons attached to objects that indicate their changed state. Because there are many types of icons, we have developed and are testing the effectiveness of three different sets of change indicators.

1. Rudimentary graphical indicators use simple symbols to represent changes + for addition, - for deletion, and ▲ for modifications.

2. Change icons often seen in today's systems, with 🗋 for addition (the blank document often represents 'new', 💣 for deletion, and ⚠ for modification.

3. Text-based icons: Add for addition, Del for deletion, and Mod for modification.

In the following three illustrations, these representations will be shown in a sample software project specified as a UML class diagram. The changes added to the sample shows the situation where two classes have been added to the UML diagram, and one class has been modified by having a method deleted and a data field added to it.

6. FUTURE WORK

We are currently running a study to evaluate the effectiveness of iconic change indicators in general, as well as how particular change indicators perform. We are implementing and testing these simple change management ideas by modifying an existing UML editor. Of course, iconic change indicators are just scratching the surface of how to represent changes, and we expect to develop other much more radical methods for representing change as well as for filtering changes.

References

[1] Bohner, S. and Arnold, R. An Introduction To Software Change Impact Analysis. Software Change Impact Analysis. IEEE Computer Society Press, Los Alamitos, CA 1996.

[2] Dellen, B. Change Impact Analysis Support For Software Development Processes. A Ph.D. Thesis from the University of Kaiserslautern 2000.

[3] Parsowith, S., Fitzpatrick, G., Kaplan, S. and Segall, B. (1998) Tickertape: Notification and Communication in a Single Line. In *Proceedings 3rd Asia Pacific Computer Human Interaction (APCHI98)*, Japan, IEEE Computer Society, pp 139-144.

[4] Robbins S. An open source UML editor (v 0.5.2) from the University of California, Irving. Irving, California Available online at http://argouml.tigris.org

[5] Neuwirth, C., Chandhok, R., Kaufer, D., Erion, P., Morris, J. and Miller, D. (1992). Flexible Diff-ing In A Collaborative Writing System. Proceedings of the ACM '92 (Toronto ON, 1992) ACM Press, 147 - 154.

[6] Hill, W. and Hollan, J.. Edit Wear And Read Wear. Proceedings of CHI '92 (Monterey CA, 1992) ACM Press, 3 - 10.

[7] Gutwin, C. Workspace Awareness In Real-Time Distributed GroupWare. A Ph.D. Thesis from the University of Calgary, 1997.

1a. Change Indicators

Operation	*Change Indicator Used*
Addition	+
Deletion	-
Modification	▲

1b. Example

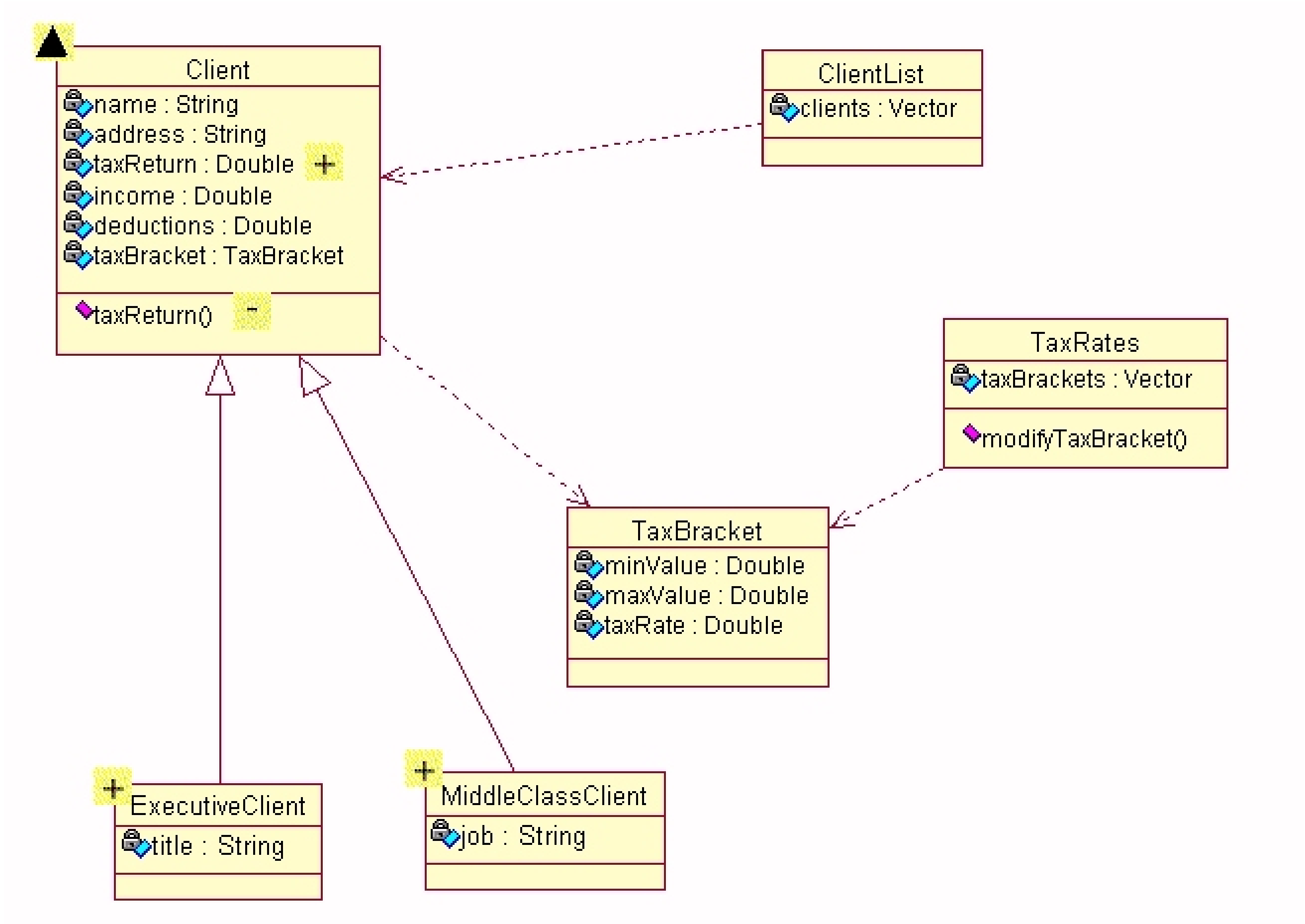


Figure 1. Rudimentary Graphical Indicators

2a. Change Indicators

Operation	Change Indicator Used
Addition	🗋
Deletion	💣
Modification	⚠

2b. Example

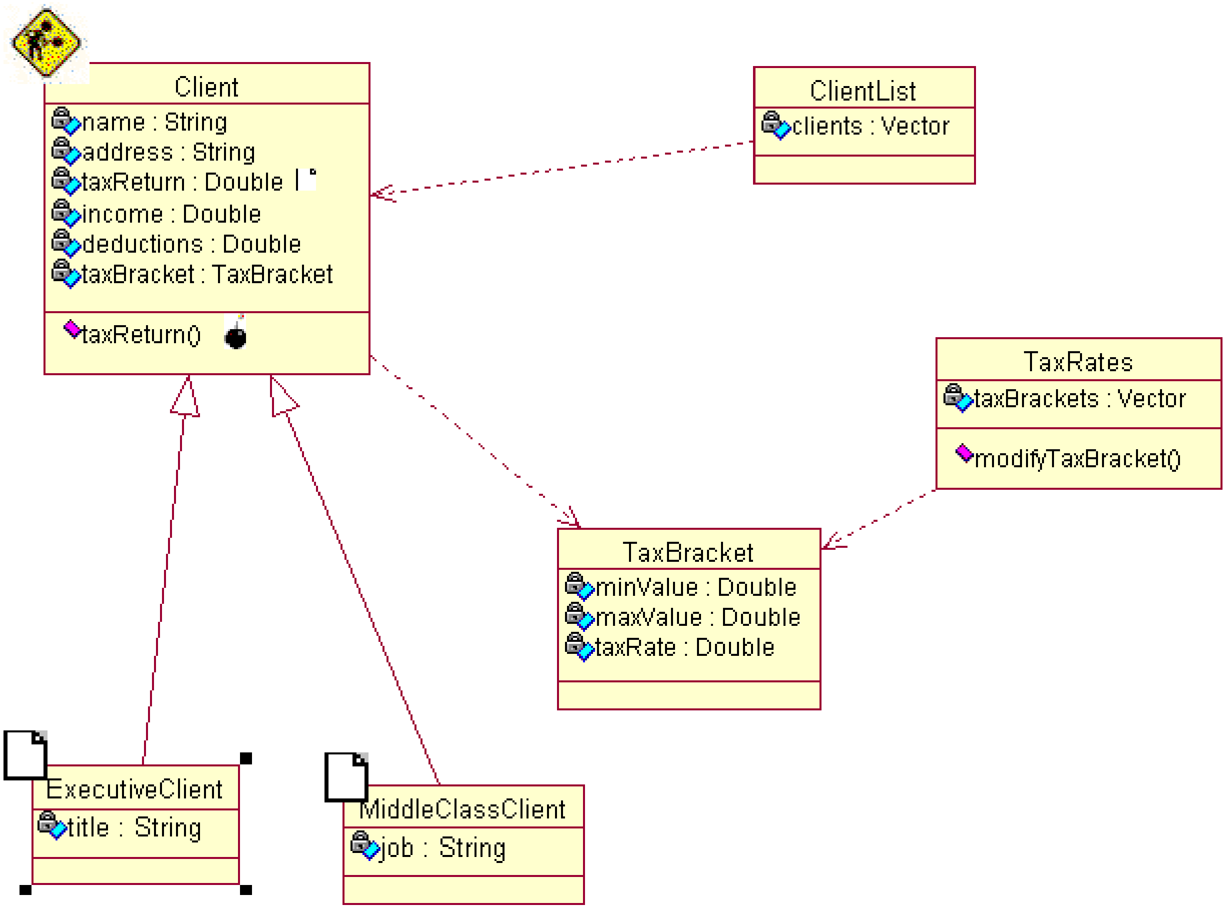


Figure 2. Common software based icons as change indicators

3a. Change Indicators

Operation	Change Indicator
Addition	Add
Deletion	Del
Modification	Mod

3b. Example

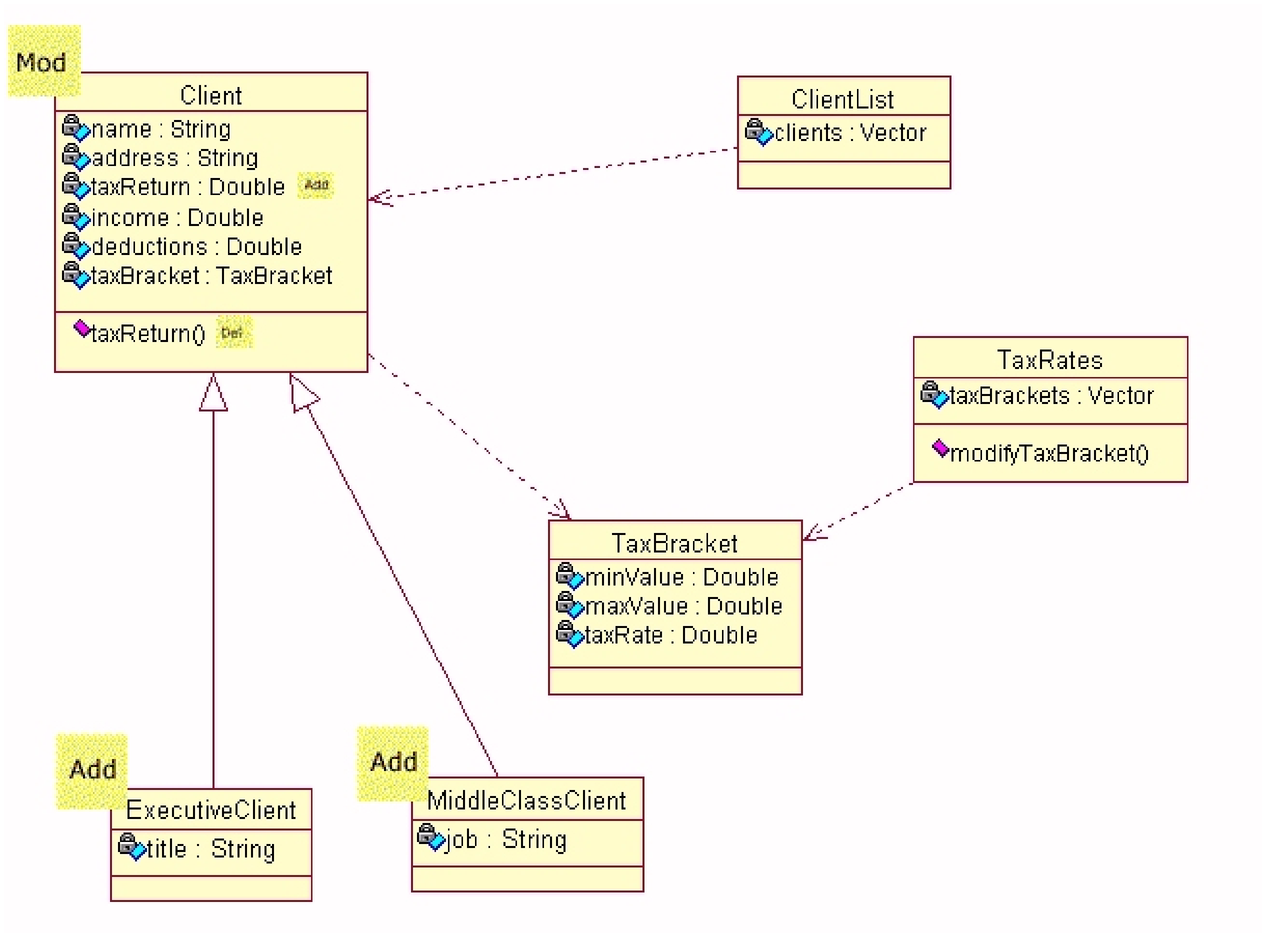


Figure 3. Text-based change indicators

A Single Display Groupware Widget Set

Ana Zanella
Department of Computer Science
University of Calgary
2500 University Dr NW
T2N 1N4
+1 403 220-7686
azanella@cpsc.ucalgary.ca

Saul Greenberg
Department of Computer Science
University of Calgary
2500 University Dr NW
T2N 1N4
+1 403 220-6087
saul@cpsc.ucalgary.ca

ABSTRACT

Our research concerns the design of a widget set that directly supports Single Display Groupware (SDG) applications that have Multiple Input Devices (MID). In this paper, we present some of the issues related to SDG development and describe why existing widgets and interface components are not appropriate to SDG application. Because this work is still in its early stages, no solutions are presented in this paper. Rather, we present a research proposal that describes the specific problems we will be dealing with as well as initial steps to solving the problem.

Keywords

Widget design, single display groupware, multiple input devices, computer supported cooperative work

1. INTRODUCTION

Single Display Groupware is a class of applications that supports multiple simultaneous users interacting in the same room on a single shared display, ideally using multiple input devices [24]. While SDG is not new — we see SDG systems in commercial use for entertainment, education, and software engineering — it is still only lightly explored as a research area within Computer Supported Cooperative Work (CSCW). Instead, CSCW researchers concentrate on supporting people working together from different locations.

SDG deals with applications in which people are sharing the same screen and therefore the same visual screen components with other people in the environment. Of course, we can do this now with conventional computers: people can sit around a display (whether a large screen or projected image) and use a single mouse and a keyboard to input data to the computer. The problem is that the input device has to be shared, and people have to take turns, perhaps even moving around and changing places. This can be awkward and difficult when people are trying to cooperate and interact with the system. For example, researchers have noted that children in a group who are playing a game on a shared computer may sometimes resist sharing by hogging the input device [6] [12] [24].

Sharing single input devices has more subtle aspects to it. For example, it can change the notion of 'who is in control'. Some researchers have observed that when several people are working on the same computer, the person who is operating the input device is not the person in charge [14]. Often, what happens is that this person usually follows orders, where they perform activities requested by people seated close by. Consequently, when the person operating the input device is executing a task on their own it is because others have lost interest in it and are doing something else [24].

The basic problem is that conventional computers and operating systems have been designed to give support to only one user on each machine. Small screens are difficult to gather around [9]; the single mouse and keyboard are passed around only with effort; simultaneous input is almost impossible. At the systems and hardware level, there is little or no support given to co-located people who share a display.

What makes SDG truly different from the above situation is that:

- every person has their own input device that they can use to interact with the application:
- the software is aware of this: it knows who is doing what, and responds accordingly.

As examples there are several systems that provide a large electronic shared display (e.g., the size of a whiteboard) that people can use to review software design solutions: different members can write down notes, brainstorm ideas, or modify the design simultaneously using multiple input devices [23] [19] [7] [14]. Similarly, two or more children can simultaneously play a game on the same computer using several mice or joystick (this is now done with many specialized computer hardware devices for games, such as Nintendo 64), edit a story or a draw a picture together [24].

From a human factors point of view, there is nothing unusual about this situation. Tang, for example, noticed that people sharing visual displays such as tabletops and whiteboards often gesture and make marks simultaneously over the surface [26][27]. The 'multiple input devices' in this case are, of course, people's hands and their pens.

In summary, the main objective of SDG applications is to support the way people work on the same computer at the same time. SDG should make it easy for people to share information, and to collaborate over a task. It should let people interact in a simple and effective way, which includes simultaneous input. To do this, SDG must provide multiple input devices (MID), as well as interaction techniques (or widgets) that afford simultaneous input. Before exploring how this can be done, we will first provide some background on SDG applications (Section 2). Section 3 will list several issues that appear when specifying and implementing SDG. We then rephrase the problem as a research area, and close by describing our initial forays into it.

2. TYPES OF SDG APPLICATIONS

Research in SDG has been performed in areas such as children's applications, meeting tools, shared editors and multiple input device systems.

Perhaps the largest body of SDG research is on systems supporting interacting children, typically educational applications or games and entertainment applications [1] [6] [12] [3] [4] [24] [25] [13]. For example, educational editors enable children's cooperation by supporting drawing while games often encourage competitive rather than cooperative interaction. Various studies found that children using a single mouse application tend to fight for the control of the device, where the one not controlling the mouse usually gives orders to his or her partner regarding tasks to be executed [24] [25] [12] [6] [3] [4]. However, holding the input device affects the feeling of involvement. When one child stays without the input device, he or she shows more off-task behavior than the child holding the device.

Tang studied how people use traditional shared displays such whiteboards and tabletops. He observed that small groups working co-located gesture simultaneously the majority of time [26] [27]. Also informal talking and face expressions are other important actions observed in the process. Gesturing, talking and observing other's expressions are difficult actions to support in computer systems.

Stefik [23] and Tatar [28] studied the use of multiple and single displays in meeting systems. The studies show that when people are working together in the same room and sharing a single output device they concentrate more on the task than when the users have their own separate display. Even when users have to take turns to input data, sharing a single display enhances the collaboration.

SDG editors allow co-located people to edit data simultaneously through multiple input devices [11] [14] [16] [20] [21]. For example, MMM (Multi-Device, Multi-User, Multi-Editor) [2] was one of the first SDG environments. It emphasized the use of up to three mice, where people could simultaneously interact with the editor and input data. The users shared a single display. MMM supported editing of text and rectangles and used object colors to distinguish between different users.

Specific support for formal meetings activities is another focus of research on SDG. These applications emphasize the idea of people creating some material or reviewing shared information via structured or semi-structured process[8] [9] [11]. Examples include software diagrams and brainstorm sessions. Davis [5], for example, presents a system that maintains notes taken during a meeting using Palm Pilots, which are then stored and made available to others through a WWW based repository.

Research on software supporting multiple input devices is based on hardware and architecture issues for connecting extra input devices to a system, such as multiple mice, keyboards or personal digital assistants (for example, Palm Pilots and Windows CE Devices). Robertson uses Palm Pilots to control a TV[22]. Myers used Palm Pilots as a input device to aid the user in repetitive activities such as scrolling the screen or controlling PowerPoint presentations [17] [18]. Studies show that providing multiple input devices enhances the collaboration and the interaction among the users [1] [16] [17] [2] [24] [12] [11]. However, in most of this systems provide one type of input device, even though all agree that having different types of devices could be even more beneficial to the whole cooperative process. This is because in different situations one type of device can be more appropriate than other. For example, a keyboard is more appropriate than a mouse when there is massive text input on the application; using a PDA the user has an extra and private display and can maintain private information.

3. SOME TECHNICAL ISSUES IN SDG DESIGN AND IMPLEMENTATION

Several technical issues appear when specifying and implementing SDG applications with multiple input devices. These include: identifying users' devices and users' work, providing widgets that recognize and respond to actions of different users; providing interface elements that avoid interference in others' work and displaying the global status of all users of the system. A secondary problem is how to adapt conventional groupware applications so they can support SDG and multiple input devices.

3.1 Identifying users' device and users' work

SDG assumes that every person has his or her own input device. An obvious implication is that the system should be able to support multiple devices, that the software must be able to identify individual devices, and that (at some level) a user can be associated with particular devices and its input data. Another fundamental requirement is that the system has to control and handle simultaneous inputs by several users. While some systems get around this by insisting on turn taking, we believe it would inhibit natural social interaction that includes simultaneity [26] [27].

While there are systems that use a homogeneous set of input devices (e.g., multiple mice [1]) we expect that an effective SDG system will allow different people to use different input devices. Each person may have at their disposal different combinations of mice, keyboard, joystick, and even a personal digital assistant (such as a Palm Pilot or a Windows CE machine). Consequently, the system should be designed to handle and support heterogeneous devices working concurrently. There have been some attempts to address this: some SDG systems support several mice [2]; others enable input to be done using Palm Pilots [14]. We are aware of only one system that allows both mouse and PDA [17].

3.2 Providing widgets that recognize and respond to actions of different users.

In SDG applications users are not only sharing the same display but the same interface components as well. This may include widgets such as menus, buttons, dialog boxes, etc.

Widgets in single user applications can be selected only one at a time. It is not possible to select the drawing tool and text tool at the same time. Also, some widgets can only be accessed in a certain order; for example, paste is only possible after copying or cutting actions. Besides, some buttons can not be pressed in some situations, such as, changing the font type if the user is drawing.

This is a problem in SDG because individual users may be in different modes and using different functions of the application at the same time. For example, multiple users of an SDG drawing editor who choose different tools, colors and styles as they work will be in different drawing modes. One user may want to be in a mode where he can draw a red circle, while another may want to write text using a different color. Problem can also happen if one user selects and copies a part of the drawing and other user selects the button to paste it, so the widgets selected must correctly identify which user has selected it and respond accordly..

In this sense, the widgets must be designed for several users and the system has to identify and handle the mode each user is in, for example drawing a line in red, inputting text in black, adding a comment in green, suggesting a change to the information, etc. Existing tool palettes don't support the multiple modes required, because there is only one 'global' notion of mode. Similarly, they cannot provide feedback that shows how different people are in different modes. Thus new interface techniques must be developed for SDG applications [2]. While these may be inspired by conventional interface widgets, we expect that they will have to be completely redesigned.

3.3 Designing displays to avoid user's interference

Another problem related to SDG interfaces is that several types of interface widgets can, when used by one person, create interference in another person's work. If one person raises a menu, for example, another person is blocked from seeing or working on whatever is behind it. Similarly, dialog boxes (especially modal ones) may block others from continuing their work. Thus the use of some classical interface widgets can be distracting and confusing during collaborative work,. These too must be redesigned (perhaps quite radically) to make them work within SDG applications.

3.4 Providing awareness of the other users' work

People should be able to maintain awareness of other users' status, such as their current mode and preferences, in order to better understand the overall collaborating process. Having every user in the same room is not enough to understand the cooperation and the involvement of each user in the specific activity being performed; so designing the system to handle this can improve the users awareness of the task.

3.5 Redesigning conventional applications

Another important issue is related to the significant changes that have to be done in conventional applications in order to support SDG. For example, we would have to add functionality to a drawing editor for multiple selection of drawing tools and different drawing modes. In a presentation system, it should be possible for every user to annotate slides simultaneously, to identify each user's annotations, and so on.

Innovative interface techniques must be developed for MID based SDG applications. While these may be inspired by conventional interface widgets and conventional CSCW interaction model, we expect that they will have to be significantly redesigned.

4. AREA OF RESEARCH

The main objective of my research is to design and develop MID-based SDG that naturally and effectively support people in both their individual and collective work. Specifically, we will define and implement a set of interface techniques (or widgets) that are appropriate to SDG applications. At its most basic, the task scenario for this design includes a group of people working together on a task, where the task will involve them moving between their individual and collaborative group work as warranted by the situation.

This is still early work, and much has to be done to develop the detailed methodology necessary to achieve this objective. However, there are three obvious steps that must be done to begin and carry through this work.

1. Build or use an already developed architecture that supports multiple input devices;
2. Build a device-aware widget set that addresses the human factors of SDG and that can be used to rapidly prototype SDG systems;
3. Evaluate various widgets by studying how people use these widgets and multiple input devices in different kinds of applications, and how they affect the collaborative process as a whole.

My primary research interest is in the development of device-aware widgets, where I expect to propose and investigate different widget designs based on real user needs. However, the first topic is important, as it will provide the architecture necessary for rapidly building different widget designs. The second and third topics are very related to each other, because the evaluation of the widgets will be made observing people using the widgets. Problems reported by the users and feedback about the usage will be used to improve the widgets.

4.1 Architecture to support multiple input devices

A serious technical problem is that conventional operating systems and applications only supply a single input device, which is inappropriate for SDG. The solution is to create or use an already developed architecture that supports multiple input devices, which can give better support for SDG applications development.

Research goals are to define how different input devices should communicate to and from the SDG system, how the SDG system should deal with these devices, and how we can encapsulate any solutions into the architecture. The overall goal is that the architecture should make it easier to build general SDG applications.

The architecture should have the following characteristics:

- Appropriate support of different kinds of input devices such as mice, PDA's, touch screen, etc. to work together simultaneously in the same application;
- Effective interaction between several input devices and the SDG system, and vice-versa;
- Easy to program API for SDG development.

Fortunately, there has already been some work in this area [2] [11] [16]. Because architecture is a secondary interest in my research, I will evaluate existing platforms. We will develop or modify an architecture from scratch only if existing platforms are inadequate for our research purposes.

4.2 Widget Set

The second part of my research will concentrate on developing a widget set that support multiple input devices, that are device aware, and that supports expected human behavior over a collaborative space. Some of the aspects that should be addressed are:

- Recognizing different types of input devices in the system;
- Assigning input devices to a user in the system and managing them;
- Appropriate management of widgets and interface components such as menus, icons, shortcuts, etc. so users do not interfere in others' work and cooperate in a natural way;
- Managing simultaneous input by several users and devices;
- Dealing with personal and public information in the system [10];
- Most importantly, supporting people's typical behaviors when collaborating together over a shared visual surface.

This is the most important part of the research, and several design solutions should be used to develop the widgets. The developed widgets should be tested during the next phase of my research, based on the evaluation of people's behavior using the widgets.

4.3 Study the behavior of people when using multiple input device applications

Evaluating widgets is crucial for understanding design deficiencies, as well as for knowing if we have developed effective solutions. We will study how people behave when they use SDG and multiple input device applications, implemented using the device aware widgets described in the previous section. We will look for patterns of collaboration, problems related to SDG application use, and different kinds of reaction people have to them. We will compare the use of classical systems and SDG systems to see how people change their behaviors, their collaboration and their interaction.

The main objective is to test the widget developed, and to find out people's behavior when interacting with each other using different kinds of SDG. Observing people's behavior can produce feedback for redesigning the widgets, in order to make them more natural to use. Example systems to be used by people include design review tools, editors, meeting tools and games.

The kinds of reaction/behavior we expect to observe are:

- How people collaborate using multiple input devices;
- How people interact to each other in the SDG system and outside the system, that is how much the system enhance the collaboration process;
- The quality of the collaboration and how it affects the results of the process;
- How different SDG systems affect the collaboration process.

5. CURRENT PROGRESS

Our research in Single Display Groupware is just beginning. To prepare ourselves, we are pursuing the following activities:

- Complete a literature review;
- Analyze current SDG applications in different areas;
- Analyze current SDG architectures;
- Develop Windows CE programming skills in order to program CE devices to be used as input device;
- Develop device-aware widgets for SDG applications;
- Identify SDG applications that use multiple input devices, and build them using a SDG architecture;
- Test SDG applications that use the developed widgets, in order to analyze people's behavior.

6. CONCLUSIONS

Collaboration benefits when participants are co-located, as opposed to distributed, because people tend to gesture and talk more, face and body expressions are easily percept by others, and people tend to interact more since they can see each other. Existing hardware and software systems were designed with individual usage in mind, and so fail to adequately support co-located collaboration.

Research studies concentrate in problems related to the interface design and the usage of multiple input devices. It is important for an SDG system to provide: an interface that can respond appropriately to several users doing different activities; identify and distinguish between different input devices; display modes that particular users are in; and, display the global status of all users of the system. In this sense, adapting conventional groupware applications to SDG requires substantial changes.

Our research will focus on developing a MID-based SDG that naturally and effectively support people interacting while engaged in co-located activities. We will define and implement a set of interface or widgets that are appropriate to SDG applications. To achieve the objectives we will define or use and already existing SDG architecture, that would allow fast building of different widget designs. We also plan to test and evaluate the widgets observing how people use them in order to refine the design solutions and provide appropriated support for SDG applications.

7. REFERENCES

[1] Bederson, B. B., Hollan, J. D., Druin, A., Stewart, J., Rogers, D., & Proft, D. (1996). Local Tools: An Alternative to Tool Palettes. In Proceedings of User Interface and Software Technology (UIST 96) ACM Press. 169-170.

[2] Bier, Eric A.; Freeman, Steve. (1991) MMM: A User Interface Architecture for Shared Editors on a Single Screen Proceedings of the fourth annual ACM Symposium on User Interface Software and Technology, November 11-13, 1991. 79-86

[3] Bricker, L., Tanimoto, S., and Hunt, E. (1998) Effects of Cooperative Interactions on Verbal Communication. Paper submitted to ACM CHI'99, Pittsburgh, PA.

[4] Bricker, L., Inkpen, K., Stewart, J., Myers, B. and Poltrock, S. (1998) Single Display Groupware: Exploring Computer Support for Co-Located Collaboration. Panel submitted to ACM CHI'99, Pittsburgh, PA.

[5] Davis, Richard C.; Brotherton, Jason A.; Landay, James A.; Price, Morgan N.; Schilit, Bill N. (1998) NotePals: Lightweight Note Taking by the Group, for the Group. UC Berkeley Computer Science Division Technical Report UCB//CSD-98-997.

[6] Druin, Allison; Stewart, Jason; Proft, David; Bederson, Ben; Hollan, Jim. (1997) KidPad: A Design Collaboration Between Children, Technologists, and Educators Proceedings of ACM CHI'97, ACM Press. 463-470

[7] Elrod, Scott, et all. (1992) Liveboard: a Large Interactive Display Supporting Group Meetings, Presentations and Remote Collaboration. Human Factors in Computing Surveys, Proceedings of ACM SSIGCHI'92. Monterey, California, 1992. 599-607.

[8] Greenberg, S. and Boyle, M. (1998) Moving Between Personal Devices and Public Displays. In Workshop on Handheld CSCW, held at ACM Conference on Computer Supported Cooperative Work, November 14.

[9] Greenberg, S. (1999). Designing Computers As Public Artifacts. International Journal of Design Computing: Special Issue on Design Computing on the Net (DCNet'99), November 30 - December 3, University of Sydney. http://www.arch.usyd.edu.au/kcdc/journal/

[10] Greenberg, S., Boyle, M. and LaBerge, J. (1999) PDAs and Shared Public Displays: Making Personal Information Public, and Public Information Personal. Personal Technologies, Vol.3, No.1, March.

[11] Hourcade, J. P.; Bederson, B. B. (1999). Architecture and Implementation of a Java Package for Multiple Input Devices (MID). Tech Report HCIL-99-08, CS-TR-4018, UMIACS-TR-99-26, Computer Science Department, University of Maryland, College Park, MD

[12] Inkpen, K., McGrenere, J., Booth, K.S., and Klawe, M. (1997). The effect of turn-taking protocols on children's learning in mouse-driven collaborative environments. Proceedings of Graphics Interface '97. 138-145.

[13] Inkpen, Kori M. (1999). Designing Handheld Technologies for Kids. Personal Technologies Journal, 3(1), May 1999

[14] Mantei, Marilyn (1988) Capturing the capture concepts: a case study in the design of computer-supported meeting environments;; *Proceedings of the conference on Computer-supported cooperative work* , 1988, Pages 257 - 270

[15] Myers, Brad A.; Stiel, Herb; Gargiulo Robert. (1998) Collaboration Using Multiple PDAs Connected to a PC, Proceedings CSCW'98: ACM Conference on Computer-Supported Cooperative Work, November 14-18, 1998, Seattle, WA. 285-294.

[16] Myers, Brad A. (1999) An Implementation Architecture to Support Single-Display Groupware. Carnegie Mellon University School of Computer Science Technical Report, no. CMU-CS-99-139 and Human Computer Interaction Institute Technical Report CMU-HCII-99-101. May, 1999

[17] Myers, B.; Lie, Kin P.; Yang, B. Two-Handed Input Using a PDA and a Mouse. Available on-line at: http://almond.srv.cs.cmu.edu/afs/cs.cmu.edu/project/pebbles/www/index.html

[18] Myers, B.; Miller, R.; Evankovich, C.; Bostwick, B. Individual Use of Hand-Held and Stationary Computers Simultaneously. Available on-line at: http://almond.srv.cs.cmu.edu/afs/cs.cmu.edu/project/pebbles/www/index.html

[19] Nunamaker, J. A.; Dennis, Alan R.; Valacich, Joseph S.; Vogel, Douglas R.; George, Joey F. (1991) Electronic Meetings Systems to Support Group Work. CACM, July, 1991. 34(7) 40-61.

[20] Rekimoto, Jun (1998) A Multiple Device Approach for Supporting Whiteboard-Based Interactions, Proceedings of ACM CHI'98. 344-351.

[21] Rekimoto, Jun.(1998) Multiple Computer User Interfaces. Cooperative Buildings, Integrating Information, Organization and Architecture, First International Workshop, CoBuild'98, Lecture Notes in Computer Science No.1370, Springer-Verlag, 1998.

[22] Robertson, Scott; Wharton, Cathleen; Ashworth, Catherine; Franzke, Marita. (1996) Dual Device User Interface Design: PDAs and Interactive Television . Proceedings of ACM CHI96.

[23] Stefik, Mark; Foster, Gregg; Bobrow, Daniel G; Kahn, Kenneth; Lanning, Stan; Suchman, Lucy. (1987) Beyond the Chalkboard: Computer Support for Collaboration and Problem Solving Meetings. Communications of the ACM, 1987, 30(1). 32-47.

[24] Stewart, Jason; Raybourn, Elaine; Bederson, Benjamin; Druin, Allison. (1998) When Two Hands are Better Than One: Enhancing Collaboration Using Single Display Groupware Extended Abstracts of ACM CHI'98, ACM Press. 287-288.

[25] Stewart, Jason; Bederson, Benjamin B.; Druin, Allison. (1999) Single Display Groupware: A Model for Co-present Collaboration, Proceedings CHI'99: ACM Conference on Human Factors in Computing Systems, May 15-20, 1999, Pittsburgh, PA. 286-293

[26] Tang, John C. (1989) Listing, Drawing and Gesturing in Design: a Study on the use of Shared Workspaces by Design Teams. Xerox PARC Report #SSL-89-3 (PhD Thesis Stanford University)

[27] Tang, John C. Findings from observational studies of collaborative work. In Computer Support Cooperative Work and Groupware. Edited by Saul Greenberg, 1991. Pages 11- 26

[28] Tatar, Deborah G.; Foster, Gregg; Bobrow, Daniel G. Design for Conversation: lessons from Cognoter. In Computer Support Cooperative Work and Groupware. Edited by Saul Greenberg, 1991. Pages 55- 79