iChat and Gabber
Interfaces Investigated

Julian Missig

05-499K Computer-Mediated Communication

6 May 2005

Table of Contents

Background

Modern instant messaging (IM) consists of two components: Presence, the ability to see and subscribe to the availability status of another user, and (Instant) Messaging, the ability to send short text-based messages delivered almost immediately to the recipient (Day, Aggarwal, Mohr, & Vincent, 2000). Instant messaging is defined in this way so as to exclude communication media like the Internet Relay Chat (IRC) protocol, which while interesting, do not quite have the same form of interactions as instant messaging, primarily because there is no definitive way to ensure that one always receives the presence for the user a person specifies—in fact, the definition of instant messaging almost requires that there be some sort of standardized addressing. The rest of this paper will discuss instant messaging with this definition in mind.

Instant messaging is popular in the workplace (Nardi, Whittaker, & Bradner, 2000), popular within academia, and popular for informal social communication. As far as technologies go, instant messaging is quite young, maybe about the age that email was in the early 1980s as it was just becoming standardized. The nature of instant messaging allows for private networks as well as large public networks (just like email). There are at least four major public instant messaging networks: AOL Instant Messenger (America Online, Inc., 2005), MSN Messenger (Microsoft Corporation, 2005), Yahoo! Messenger (Yahoo! Inc., 2005), and XMPP/Jabber (Jabber Software Foundation, 2005), which I will refer to as AIM, MSN, Yahoo, and Jabber, respectively. These four public networks

cannot communicate with one another without the use of third-party software (which is often illegal given the Terms of Service for using these networks). Email once had many separate protocols and networks, but it eventually was standardized, as is likely to happen to instant messaging some day. Because of the popularity of instant messaging and its relative age, the clients for these protocols are changing rapidly. AIM, one of the oldest clients still in common use, is having its interface completely rebuilt (see http://beta.aol.com/projects/tritonbeta/). Many people even work to make third-party clients which can communicate on these networks (for example, Trillian by Cerulean Studios, 2005). When I refer to a "typical" or "popular" instant messaging client, I am referring to the official clients for AIM, MSN, or Yahoo.

With the rapidly changing software arena in mind, it is easy to see how important, interesting, and good user interface developments could be lost. In this paper I document user interface features of two instant messaging clients, iChat and Gabber, with the hope that those features which have been implemented well will not be forgotten by future clients and will serve as a direction for future research. Rather than reviewing the related Computer-Mediated Communication and Human-Computer Interaction research and then discussing the features I wish to document, I have organized this paper around the interface features I document, and discuss the relevant research with each feature.

iChat AV

iChat AV was created by Apple Computer, Inc. (2005). The initial development began in 2000 (Alfke, 2004), with a first release to the general public in 2002 as part of Mac OS X 10.2. The version covered by this paper is iChat AV 3.0, which is included with Mac OS X 10.4 Tiger. iChat communicates via the AIM and Jabber networks, as well as supporting local network messaging using Bonjour (formerly known as Rendezvous). I was not involved in the creation of iChat, nor have I specifically interviewed iChat developers for this paper, so it should be understood that any reasoning put forth for features is my own, and not the reasoning of Apple Computer, Inc.

Gabber

Gabber started as a collaboration between Dave Smith and myself in 2000. It is a client for the Gnome Desktop platform for Linux (GNOME Project, 2003). The first major version, which I refer to as Gabber 1 (Missig & Smith, 2002), was created for Gnome 1.2. When Gnome 2.0 was released, Gabber had to be almost completely rewritten. This work was done primarily by Thomas Muldowney and myself. I refer to this second major version as Gabber 2 (Missig, 2004), although it never became as complete as Gabber 1 was. Many other people contributed time and energy in

Gabber 1, version 0.8.8.

producing Gabber, but I was the primary person in charge of the user interface, so I can speak for its reasoning.

Affordances of Instant Messaging

"Grounding" in communication is the collaborative act of both (or all) parties to build up a common set of actions and terminology with which information, goals, and ideas can be communicated. Clark and Brennan (1991) listed eight main constraints of grounding which differ with different forms of media. Because of the great utility in describing constraints on communication using these affordances, I repeat Susan Fussell's summaries here.

1. **Co-Presence**. People share the physical environment, including a view of what each other is doing and looking at. IM does not support this, allowing users to multitask while communicating.
2. **Visibility**. People can see one another but not what each is doing or looking at. IM does not support this (without video chat or webcams), allowing users to multitask while communicating.
3. **Audibility**. People can hear one another. IM does not support this (without audio chat).
4. **Cotemporality**. Messages are received close to the time that they are produced, permitting fine-grained interactivity. IM conversations are typically informal, with immediate responses not always required.

5. **Simultaneity**. Multiple people can send and receive messages at the same time, allowing backchannel communication. IM does not usually feature this.
6. **Sequentiality**. People take turns in an orderly fashion in a single conversation. IM leads to topics often being overlapped and interleaved.
7. **Reviewability**. Messages do not fade over time. IM allows for this at least as long as a window persists.
8. **Revisability**. Messages can be revised before being sent. IM partially supports this, with a small text area in which people can edit messages.

iChat AV

As mentioned before, I cannot directly speak for the reasons behind the designs of iChat, but I have been using it for a long time—and I have experience as an IM client developer. It is often assumed that all of the extra aesthetics in iChat add nothing to the actual interaction experience. This assumption is not accurate: Some of the aesthetics do have good reasons, even from the perspective of Computer-Mediated Communication.

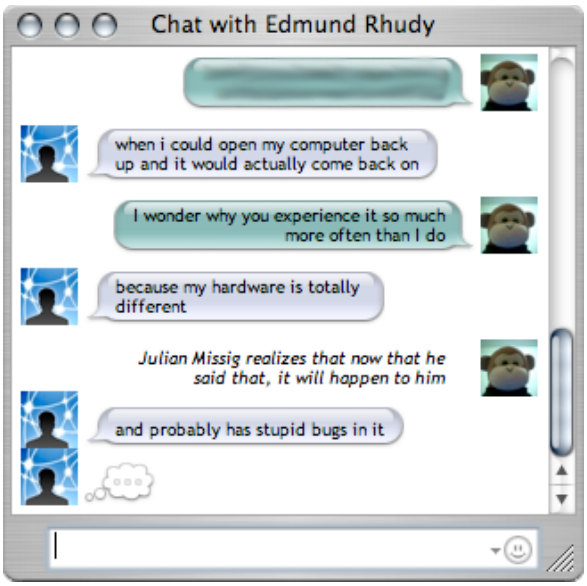
iChat's main concern appears to be conveying identity. It goes out of its way to provide multiple cues as to who it is you are talking to, and who it is on your list. iChat tries to convey people as people, not just names on a list.

Speech Bubbles

Gergle, Millen, Kraut, and Fussell (2004) investigated how the amount of displayed text in a chat window affects conversations during task performance. They concluded that more visible history is better—in part because conversations do not need to be re-grounded, there is immediate context as to what was just said. Gergle et al. (2004) go on to claim that iChat's interface may be bad because a lot of space is spent on avatars and speech bubbles rather than on raw conversation text.

iChat's chat windows, however, do not have extra toolbar buttons, a menu, or even more than a single line of input area visible by default—of note is the fact that the input area grows to fit the amount of text the user has written (by growing the window, rather than shrinking the conversation display area), thus allowing for greater

revisability as described by Clark and Brennan (1991). The window is almost entirely conversation. The window is also easily resizable to allow for even more conversation to be displayed. The amount of conversation displayed is not actually that much less than typical IM clients because of the sheer simplicity of having nothing else in the window.



iChat window with speech bubbles

The speech bubbles elegantly differentiate between sender and receiver—the left alignment versus right alignment, avatars on opposite sides, different subtle background colors, and different fonts all add up to a salient differentiation between the two. It is relatively easy to scan the conversation seeing who said what. It is rather common to scan back in history looking for something the other person said—this differentiation makes that very easy. Vronay, Smith, and Drucker (1999) identified *lack of recognition* as one of the major problems in text-based chatting. iChat's speech bubbles help mitigate this problem.

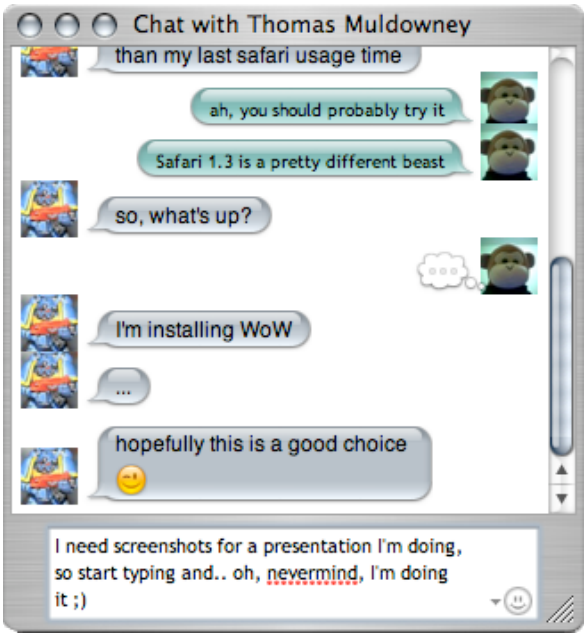
Thought Bubbles

Vronay et al. (1999) identified many factors which influence chat efficiency in a negative manner. These factors seem to stem from the affordances (or lack thereof) identified by Clark and Brennan (1991). The main problems they identified were:

1. **Lack of recognition**. Users find it difficult to associate a nickname with what that speaker has said in the past if the users do not know the speaker.
2. **Lack of intention indicators**. It is difficult or impossible to tell if a particular comment was directed at a particular user or conversation turn.
3. **Typing inefficiency**. Most people type slower than they talk, which can often be misinterpreted as silence in a conversation.
4. **Lack of status information**. There is no way to tell if a user is actively participating or is no longer in front of the computer.
5. **Lack of context**. There is no way to tell what happened in earlier conversations or what was taking place in the conversation before the user arrived.
6. **High signal-to-noise ratio**. I assume Vronay et al. (1999) meant *Low* signal-to-noise ratio. They cite the fact that most participants do not know one another as causing a high amount of introductory socialization which does not contribute to the conversation.
7. **General uselessness of the chat history**. Although many clients save the entire history of chats, those saved histories are very rarely used, if at all.

Vronay et al.'s (1999) studies focused around the idea that a lot of these problems come from the loss of timing-specific information—it is difficult to tell at which point in the conversation someone started typing. Because this information is lost, it can be unclear which specific message a person is responding to, especially if several questions have been asked and the person responds with "Yes." Vronay et al.'s (1999) work contains a good list of issues, and their experimental clients had a wide variety of ways to convey additional status and context. They did not, however, settle upon a definitive answer as to how one can convey more timing information than typical IM clients currently do, which typically involves a simple binary status display: User is composing a message or user is not current composing a message.

iChat has a solution which is at least incrementally better than typical IM clients in the direction Vronay et al. (1999) were heading. When a user begins typing, iChat displays a thought bubble coming out of that user's avatar. The thought bubble maintains that position in the conversation, even as the other person sends more messages. This indicates exactly where in the conversation a user began typing without requiring any extra information parsing on either side.



iChat chat window with a thought bubble indicating start of reply

However, once a user finishes typing, the thought bubble disappears and the message appears at the end of the conversation (as any new message would). This does not allow either participant of the conversation to later figure out where a message began being composed, so it does not completely solve the problems Vronay et al. (1999) were looking at, but it is a good step forward.

New Message Interruption

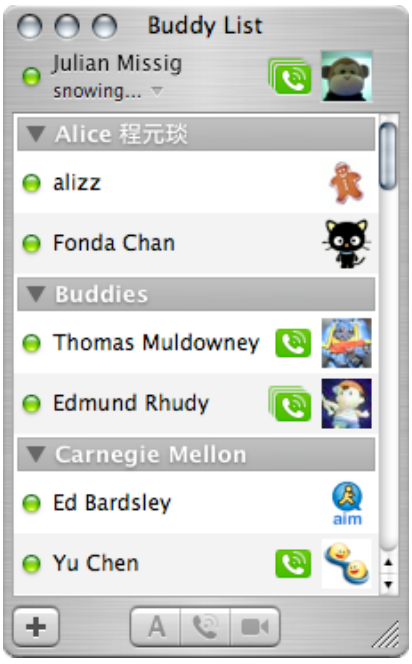
While on the topic of new messages appearing, it is noteworthy that iChat uses a smooth upward scrolling effect to move all previous messages when a new one appears. McCrickard, Catrambone, Chewar, and Stasko (2003) investigated several possible animation methods for the presentation of new information while participants were working on a web-browsing research task. McCrickard et al. (2003) found that a smooth animated scrolling effect was most successful in trading off between noticeability and distractibility. In other words, participants were aware that new information had arrived, but were not distracted by this fact as often as they were by other animation methods. This is exactly what one would want to happen when a new message arrives in an open window—iChat's smooth scrolling animation balances between notification and distraction fairly well.

Avatars

People are experts at recognizing faces (Gauthier, Tarr, Anderson, Skudlarski, & Gore, 1999). In fact, there is evidence that people become experts at categorizing objects with

complex features (Haxby, Gobbino, Furey, Ishai, Schouten, & Pietrini, 2001). This expertise means that people can very quickly associate a picture with a person (even if it is not a face), as that is essentially what is being done with faces. Walther, Slovacek, & Tidwell (2001) demonstrated that teams perform better when picture representations of the team members are used. This is true of teams that already know one another and short-term teams that do not know one another. Interestingly, it is not true of long-term teams that do not know one another, but the benefits of having picture representations for people that participants know are significant.

Apple works to promote the use of avatars (icons representing oneself in a virtual world) as one's own "face." This is consistent with iChat's attempt to build up an identity for each and every person on one's contact list—a comparatively large amount of space is devoted to each person, and each person's individual avatar appears next to their name.



iChat contact list with avatars

Avatars appear in the contact list and appear in chats. They are always there so users learn to recognize them and can learn the association between an avatar and a person. Once a person has learned an avatar, this expertise makes use of the human ability to recognize faces (as shown in Haxby et al., 2001)—users can identify their buddies by avatar

without having to attend to the actual textual name representing their friend. Making use of natural human expertise in this fashion helps to solve Vronay et al.'s (1999) *lack of recognition*.

iChat furthers this notion of representing identity by presenting the current user with a "mirror" of themself. At the top of the contact list is a "mirror." It shows the user's name, their current status, their current audio / video status (clicking on this icon with a video camera plugged in brings up a literal mirror in video form), and their current avatar. All the user has to do is click the current avatar to select a new avatar. iChat even goes to the length of keeping 16 recent avatars and providing an "Edit Picture..." dialog to allow for avatar capturing by using a video camera (encouraging the use of real faces or real-life objects) and basic cropping. Users do not even need to pull open another graphics editing program for any of that.

Simplifying Presence

Another study of instant messaging in the workplace which compared "heavy" IM users with "light" IM users (Isaacs, Walendowski, Whittaker, Schiano, & Kamm, 2002) demonstrated that light users were statistically more likely to send more messages which were not responded to (because the recipient was not around or able to respond) than heavy users—indicating that heavy users had learned to make use of availability information presented by the IM system. As per the original definition cited for instant messaging (Day et al., 2000), availability information (presence) is an integral part of IM.

iChat tries to make presence as easy and obvious as possible. The presence-setting control is built into the main contact list window, with a drop-down menu allowing users to select from previous saved status messages. The status messages of other users are displayed *right in the contact list*, under each user. This allows for a quick awareness of the user's current status, working against Vronay et al.'s (1999) *lack of status information*.

Turning "Video Conference" into "Video Chat"

Being aware of the fact that instant messaging is not the best way to perform all kinds of communication (as per Clark and Brennan's (1991) affordances), iChat has video chatting and audio chatting features. In fact, it makes it just a single click. Once each user has their video camera plugged in, one only has to click on the video or audio chat icon for a user. Most current video conference software is not this simple, and does not provide presence information. It makes sense to integrate video chatting into a contact list as iChat has done, as it is consistent with solving Vronay et al.'s (1999) *lack of status information*—being aware that video or audio chat is an option is the first step toward making use of it.

Gabber

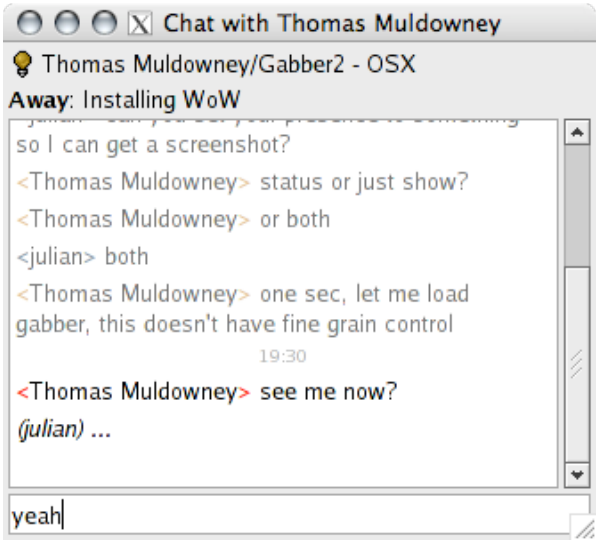
I took a lot of hints from iChat when designing Gabber 2. We implemented typing notification displays similar to iChat's thought bubbles, elegant timestamps which do not get in the way (just like iChat's), a growing text input (although ours subtracted space from the conversation area rather than growing the window), and integrated presence selection. Even though we took a lot from iChat, we also created some new user interface features which are worth investigating.

Context While Chatting

The Gergle et al. (2004) study mentioned earlier shows that more history being visible leads to better and faster task performance. Clark and Brennan's (1991) *reviewability* affordance and Vronay et al.'s (1999) *lack of context* and *general uselessness of chat history* problems all point to an overall problem with message context. Once a window disappears, the conversation restarts when a new window opens. This can be frustrating for users who do not like keeping many windows around—they have to decide between losing conversation context and having too many windows open.

Gabber takes a step forward in solving these problems by always displaying the last few lines of chat history. Whenever you open a chat window, the last few lines of conversation are displayed, partially grayed out. This was in part inspired by Jabber's multi-user chat server components, which send the last several lines of room conversation to a new user when they join, allowing for an understanding of the

conversation that was taking place as the user joined. Gabber will display timestamps in the displayed chat history, adding the date if the last conversation took place on a previous day. Bradner, Kellogg, and Erickson (1999) implemented an experimental chat client called BABBLE with a variety of features intended to help with a feeling of *co-presence* by adding a feeling of "place", but after having people use it in the workplace they believed one of their best features was presenting messages as persistent—users saw all of the messages when they signed back in. Several other third-party IM clients have implemented this, but it has yet to make its way into the more popular clients.



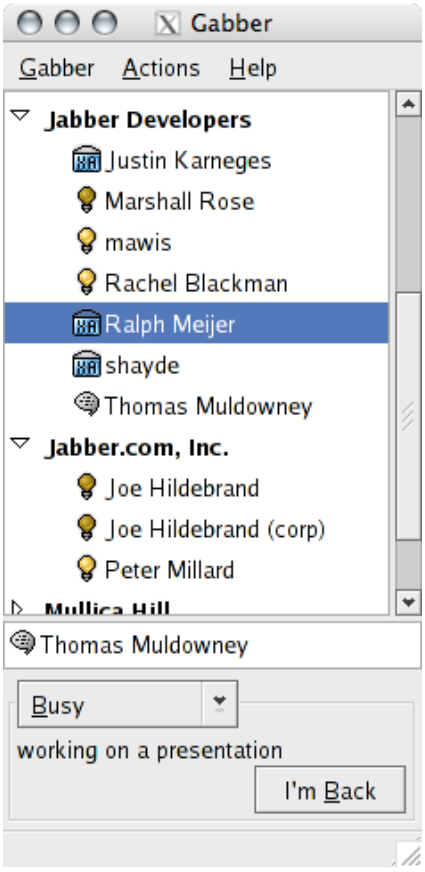
Gabber chat window with several lines of chat history

Interrupting Less

O'Conaill and Frohlich (1995) spent 29 hours shadowing two managers, and witnessed 125 interruptions. The interrupters benefited more than the recipients of the interruptions, and the recipients often failed to return to their ongoing task. Interruptions are a problem with instant messaging—the easier a medium is to communicate with, the more often people will make use of that medium and end up interrupting the party they wish to talk to.

The overall goal that I aim for, as an IM client developer, is to build a client where incoming messages are *noticeable* but not *distracting*. iChat has this working to an extent with already-open windows (smooth upward scrolling, as mentioned before), but its new conversation notifications are quite distracting. iChat pops up a partially transparent small window piece, which is better than what typical clients do (pop up a whole regular window), but is still distracting. With Gabber, I tried to come up with a way to present messages that was more aware of the current user's status.

I was vaguely aware that there was research being done on having computers detect interruptibility, and later came to read some interesting work on the topic (Fogarty, Hudson, & Lai, 2004), but for the time being I tried to find a simpler method. When users make use of the presence features of instant messaging clients to set themselves as "Away" or "Busy," they are already telling the computer that they cannot be interrupted—most instant messaging clients simply ignore that or present users with yet another annoying window! With the help of others, I came to create Gabber 2's message queuing system. By default, new conversation messages will pop up new windows as usual when a user has set him- or



Gabber window with a new chat message waiting

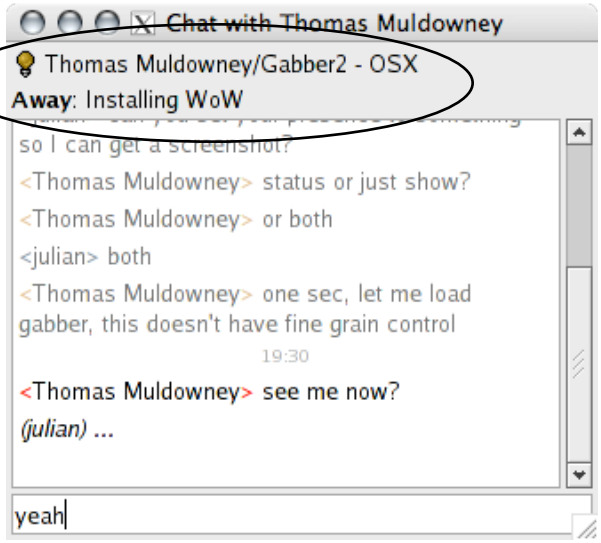
herself as "Available," but when set to "Away" or "Busy," any new conversation messages will be queued up in Gabber's contact list window. They are visible as a list, separated by message sender. A user can double-click on a name to see messages coming in from that sender. The message icons do not flash there, nor do they flash in Gabber's status docklet. They are solid icons that sit there, awaiting a time when the user is not busy. The user can choose to deal with the messages when they wish, or when they click "I'm Back" all of the messages will appear. Users have the preference of making this system always the case regardless of availability status, but anecdotally, many found it nice to actually have different functionality when "Away" or "Busy"—they found themselves using it much more often than they ever had before.

Communicating Presence While Chatting

Gabber furthers iChat's simplification of presence by providing a quick "I'm Back" button in the contact list (see above). This causes more screen space to be taken up by the presence area, better indicating to users that they are currently not Available. If the users wish to see more of their contact list at once, Gabber provides this subtle cue that the user should return to being Available.

Gabber also provides several presence states other than "Available" and "Away"—it has "Busy," "Extended Away," and "Free for Chat." "Free for Chat" is not particularly useful, but providing additional granularity as to *why* a user is away further helps with quick identification of status and when a response can be expected.

The major advance Gabber made, however, is displaying the speaker's presence in the chat window. Typical instant messaging clients only provide presence information as "messages" in the chat window—thus, if a user wishes to figure out the current presence of the speaker, they must scroll through the history to find the last status message, or they must open their contact list. Gabber provides the current status information at the top of the chat window. iChat only provides this information if one chooses "Show Chat Participants," which adds a rather large drawer to the side of the chat window.



Chat window display Thomas's presence. He is "Away" with a status message.

When a speaker is "Away," Busy," or "Extended Away," Gabber takes up an extra line of chat window space to display the speaker's status message. When a speaker is "Available" or "Free for Chat," no additional message is displayed, but a user can place their mouse over the speaker's name to see the message in a tooltip. The lightbulb presence indicator will also indicate the speaker's general availability.

Integrated presence in the chat window is a very important step toward solving Vronay et al.'s (1999) *lack of status information*. It is this feature which I most wish other IM clients (including iChat) had.

Future Considerations

While iChat and Gabber make great strides in furthering the interfaces of instant messaging, their are some obvious future improvements both clients could make other than simply integrating features from one another.

Better Chat History

Once a message scrolls out of a window, it is difficult for users to use. This is conveyed in Vronay et al.'s (1999) *general uselessness of the chat history*. Gabber does have an integrated chat history viewer, but it really is not as useful as it could be. Future instant messaging clients need to look into this and further develop solutions.

Unobtrusive Message Broadcasting

Bradner et al.'s (1999) BABBLE system supported unobtrusive message broadcasting to multiple parties. Some day the Jabber system may gain support for this, and it is important that clients look into it. This would allow for further use of instant messaging where currently one must use email.

Intelligent Interruptions

Future clients will hopefully be able to be even smarter than Gabber at determining when to interrupt with a new message, thanks to work like Fogarty et al. (2004).

User Profiles

Both Gabber and iChat lack decent interfaces for viewing and setting user profiles, which often are used as short message areas similar to an extended form of presence.

Summary

iChat works to convey that the words you see on your screen represent messages coming from real people. It dedicates screen space and many contextual cues to this task. The more cues that are visible, the better. Avatars, speech bubbles, color, font, and large spaces for names all point at the identity of the speaker.

Gabber works on bringing users into the context of the conversation. It presents past messages and current speaker presence right in the chat window. When a Gabber user is “Away” or “Busy” (again, allowing for a finer grain of context), Gabber respects that context by not interrupting the user with new messages.

Both of these clients are concerned with the people and conversations which take place while they are in use. It is important that future instant messaging clients continue to do so, and learn from these clients. It is also important to realize that research in Computer-Mediated Communication and Human-Computer Interaction had spoken of many of the problems which iChat and Gabber seek to deal with long before the creation of either of these clients. Future instant messaging clients should be aware of the current research so they too can work to solve the problems which researchers already know about.

References

Alfke, J. (2004). Jens's Tangled Job History. Retrieved 21 March 2005, from http://mooseyard.com/Jens/2004/04/jenss-tangled-job-history.

Apple Computer, Inc. (2005). Apple – Mac OS X – iChat AV. Retrieved 30 April 2005, from http://www.apple.com/macosx/features/ichat/.

America Online, Inc. (2005). AOL® Instant Messenger™. Client and protocol referred to as "AIM" in this paper. Retrieved 21 March 2005, from http://www.aim.com.

Bradner, E., Kellogg, W., & Erickson, T. (1999). The adoption and use of BABBLE: A field study of chat in the workplace. In *ECSCW* (pp. 139–157). Lyngby, Denmark: Center for Tele-Information.

Cerulean Studios. (2005). Cerulean Studios: Creators of Trillian and Trillian Pro Instant Messengers. Retrieved 21 March 2005, from http://www.trillian.cc/.

Clark, H. H. & Brennan, S. E. (1991). Grounding in Communication. In L. B. Resnick, J. M. Levine, & S. D. Teasley (Eds.), *Perspectives on socially shared cognition* (pp. 127–149). Washington, DC, USA: APA Press.

Cutrell, E., Czerwinski, M., & Horvitz, E. (2001). Notification, disruption and memory: effects of messaging interruptions on memory and performance. In M. Hirose (Ed.), *Proceedings of Human–Computer Interaction—Interact '01* (pp. 263–269). Tokyo, Japan: IOS Press.

Day, M., Aggarwal, S., Mohr, G., & Vincent, J. (2000). RFC 2779: Instant Messaging / Presence Protocol Requirements. Retrieved 20 March 2005, from http://www.ietf.org/rfc/rfc2779.txt.

Fogarty, J., Hudson, S.E., Lai, J. (2004). Examining the robustness of sensor-based statistical models of human interruptibility. In *Proceedings of CHI 2004* (pp. 24–29). New York: ACM Press.

Gauthier, I., Michael J.T., Anderson, A.W., Skudlarski, P., & Gore, J.C. (1999). Activation of the middle fusiform 'face area' increases with expertise in recognizing novel objects. *Nature Neuroscience*, 2, *6*, 568–578.

Gergle, D., Millen, D., Kraut, R.E., and Fussell, S.R. (2004). Persistence Matters: Making the Most of Chat in Tightly-Coupled Work. In *Proceedings of CHI 2004* (pp. 431-438). New York: ACM Press.

GNOME Project, The. (2003). GNOME: The Free Software Desktop Project. Retrieved 25 March 2005, from http://www.gnome.org/.

Haxby, J.V., Gobbini, M.I., Furey, M.L., Ishai, A., Schouten, J.L., & Pietrini, P. (2001). Distributed and overlapping representations of faces and objects in ventral temporal cortex. *Science, 293, 5539,* 2425–2430.

Herbsleb, J. D., Atkins, D. L., Boyer, D. G., Handel, M., & Finholt, T. A. (2002). I Think, therefore IM: Introducing Instant Messaging and Chat in the Workplace. In *Proceedings of the 2002 ACM Conference on Human Factors in Computing Systems (CHI 2002)* (pp. 171–178).

Isaacs, E., Walendowski, A., Whittaker, S., Schiano, D.J., & Kamm, C. (2002). The Character, Functions, and Styles of Instant Messaging in the Workplace. In *CSCW 2002*, 11–20.

Jabber Software Foundation. (2005). Jabber: Open Instant Messaging and a Whole Lot More, Powered by XMPP. See also Internet Engineering Task Force (IETF) RFCs 3920 and 3921. There is no official client. Common protocol referred to as "Jabber" in this paper. Retrieved 21 March 2005, from http://www.jabber.org/.

McCrickard D. S., Catrambone R., Chewar C. M. and Stasko J. T. (2003) Establishing tradeoffs that leverage attention for utility: Empirically evaluating information display in notification systems, *International Journal of Human-Computer Studies, 58*, 547-582.

Microsoft Corporation. (2005). MSN Messenger Version 7.0. Client and protocol referred to as "MSN" in this paper. Retrieved 21 March 2005, from http://messenger.msn.com/.

Missig, J. (2004). Gabber 2 and Jabberoo. Retrieved 21 March 2005, from http://gabber.jabberstudio.org/.

Missig, J. & Smith, D. (2002). Gabber: The GNOME Jabber Client. Retrieved 21 March 2005, from http://gabber.sourceforge.net/.

Nardi, B., Whittaker, S., Bradner, E. (2000). Interaction and Outeraction: Instant Messaging in Action. (2000). In *CSCW 2000*, 79–88.

O'Conaill, B. & Frohlich, D. (1995). Timespace in the workplace: Dealing with interruptions. In *Proceedings of CHI '95* (pp. 262–263). New York: ACM Press.

Trafton, J. G., Altmann, E. M., Brock, D. P., & Mintz, F. E. (2003). Preparing to resume an interrupted task: Effects of prospective goal encoding and retrospective rehearsal. *International Journal of Human-Computer Studies, 58*, 583-603.

Vronay, D., Smith, M., and Drucker, S. (1999). Alternative interfaces for chat. In *Proceedings of the 12th Annual ACM Symposium on User Interface Software and Technology* (UIST 99).

Walther, J. B., Slovacek, C. L., & Tidwell, L. C. (2001). Is a Picture Worth a Thousand Words? Photographic Images in Long-Term and Short-Term Computer-Mediated Communication. *Communication Research, 28, 1,* 105–134.

Yahoo! Inc. (2005). Yahoo! Messenger. Client and protocol referred to as "Yahoo" in this paper. Retrieved 21 March 2005, from http://messenger.yahoo.com/.