Field Deployment of *IMBuddy*: A Study of Privacy Control and Feedback Mechanisms for Contextual IM

Gary Hsieh, Karen P. Tang, Wai Yong Low, and Jason I. Hong

Human-Computer Interaction Institute
Carnegie Mellon University
5000 Forbes Ave, Pittsburgh, PA 15213
{garyh,kptang,wlow,jasonh}@cs.cmu.edu

Abstract. We describe the design of privacy controls and feedback mechanisms for contextual IM, an instant messaging service for disclosing contextual information. We tested our designs on *IMBuddy*, a contextual IM service we developed that discloses contextual information, including interruptibility, location, and the current window in focus (a proxy for the current task). We deployed our initial design of *IMBuddy*'s privacy mechanisms for two weeks with ten IM users. We then evaluated a redesigned version for four weeks with fifteen users. Our evaluation indicated that users found our group-level rule-based privacy control intuitive and easy to use. Furthermore, the set of feedback mechanisms provided users with a good awareness of what was disclosed.

Keywords: Contextual instant messaging, context-aware, IM, privacy.

1 Introduction

Instant messaging (IM) is a growing communication medium that is useful for both social and work purposes [1, 2]. While it functions as a multi-purpose communication medium, current commercial designs of IM provide minimal support for disclosing contextual information (such as location and work status) to other users. To address this concern, prior research have explored augmenting IM to include contextual information disclosure so that IM users can have better awareness of where other users are and what they are up to, and to improve IM as a communication media for collaboration, coordination and social interaction [3-6].

However, for contextual IM to flourish in everyday use, significant privacy concerns need to be addressed for supporting contextual information sharing. Previous work has highlighted two principles in designing for privacy: control and feedback [7, 8]. Without enough control, sensitive and private information could be disclosed to others. Without sufficient feedback, users would not know what has been disclosed, and that may prevent them from taking necessary precautions to protect their privacy. One design for privacy controls is to manage information disclosure on a case-by-case basis. The problem with this design is that users are always required to make the disclosure decision, which incurs interruption costs and prevents useful disclosures when they are busy or away. Another design of privacy control is a customizable rule-based control. A previous lab study has suggested that group-level

J. Krumm et al. (Eds.): UbiComp 2007, LNCS 4717, pp. 91–108, 2007.

rule-based controls are sufficient for contextual IM [9]; however, without actual field use, it is not clear what needs to be included in privacy controls and how much feedback is necessary to make contextual IM acceptable for general everyday use.

To explore this design space, we designed privacy controls and feedback mechanisms for *IMBuddy*, a contextual IM service that we developed. *IMBuddy* allows any AIM user to query an AOL Instant Messaging Robot (AIMBot) about three types of information: interruptibility, location, and current window in focus (a proxy for current task). Currently, users can only ask about selected AIM users who run our client software which collects and reports their contextual information.

We iterated our privacy designs based on actual field use. For the first deployment, ten participants used *IMBuddy* for two weeks. Although users felt comfortable using the first iteration of privacy controls and feedback mechanisms, they suggested additional feedbacks and improvements to the system. We then redesigned the system and deployed it to fifteen other students over the span of four weeks. We evaluated our designs focusing on the effectiveness of our control and feedback mechanisms.

This work offers two main contributions. First, we introduce a design for privacy control and feedback mechanisms for contextual IM. Our user study suggests that our feedback mechanisms provided ample information allowing our users to notice when their information was disclosed. Specifically, most users were aware when someone asked for their information in a suspicious way. During the study, our participants were comfortable with their privacy settings and discussed various scenarios where the information disclosed was both appropriate and useful. Components of our design can be easily reused for other contextual IM and can even be extended to information disclosure through other devices. Second, our design offers evidence that a rule-based group-level privacy control for contextual IM can work well in practice.

2 Related Work

With ubiquitous computing pushing to embed technologies in our everyday devices, it is becoming easier to sense and share user information (e.g. location). For example, prior work has demonstrated the benefits of contextual information disclosures for Media Space [10]. Similarly, the idea of contextual information sharing in instant messengers has also been explored, showing that these clients are helpful for sharing locale and activity information [3, 4, 6] and project related information [5].

As ubiquitous computing strives to make technology more invisible and integrated in our everyday lives, it becomes imperative to consider and design privacy mechanisms to properly managing information disclosures. Work by Belotti and Sellen has highlighted this issue, and they propose a design framework that focuses on feedback and control in ubicomp environments [7]. Drawing on prior research in Media Spaces [8], they define two important principles in designing for privacy: **control**, empowering people to stipulate what information they project and who can get hold of it**,** and **feedback**, informing people when and what information about them is being captured and to whom the information is being made available.

To inform our initial privacy designs for *IMBuddy*, we also drew upon several other guidelines. Previous work indicates the need for coarse-grained control as "users are accustomed to turning a thing off when they want its operation to stop" [11,

12]. Other work has demonstrated the importance of having abstract views of information [13], allowing for flexible and personalized replies [11, 14], and having mechanisms for controlling the quantity and fidelity of information disclosure [15].

An open question related to privacy is the usefulness of rule-based mechanisms. On one hand, work by Patil and Lai suggests that controlling privacy at a group level is sufficient for contextual IM [9]. On the other hand, Palen and Dourish argue that privacy is more than authoring rules [16], but rather an ongoing "boundary definition process" in which boundaries of disclosure, identity, and time are fluidly negotiated. In *IMBuddy*, we provide control and feedback mechanisms that utilize both of these philosophies. For example, we provide a rule-authoring interface as well as a history disclosure mechanism. We felt that since attention remains a scarce resource, using a rule-based approach can minimize interruptions and allow for useful disclosures when the user is busy or away. We also provide social translucency mechanisms to help users be more aware of what others know about them. Most importantly, we provide an evaluation of these different mechanisms, showing that they work well in practice.

The importance of feedback has been discussed extensively in prior work. Feedback is important because if users are not aware that their information is being disclosed, then they will be unable to react appropriately to potentially harmful requests. As Langheinrich points out, "in most legal systems today, no single data collection…can go unnoticed of the subject that is being monitored" [17]. Feedback can be further broken down into providing adequate history and immediate feedback as discussed in Nguyen and Mynatt's work on Privacy Mirrors [18]. Our work here presents the design and evaluation of several different feedback mechanisms.

3 Designs of Privacy Control and Feedback

We used *IMBuddy*, a contextual IM service that uses an AOL Instant Messaging Robot (AIMBot), to provide a framework for evaluating privacy control and feedback. IMBuddy answer queries about three types of contextual information: interruptibility, location, and active window. Our initial designs were based on formative evaluations with paper and interactive prototypes tested with five IM users.

3.1 Control

In this section, we discuss three aspects of *IMBuddy*'s privacy controls, namely its multiple information granularity levels, group-based controls, and convenient access.

Information Granularity. *IMBuddy* can disclose three types of information: interruptibility, location, and active window. To support multiple information abstractions levels, we created different levels of disclosure (see Table 1). The lowest disclosure level for all three information types is "none", which results in disclosing "no information available". Our design goal was to keep the controls simple and straightforward while still providing meaningful and appropriate information disclosures for our users; therefore, we focused on the simplest types of granularity controls and did not explore more complex controls based on time or location, etc.

For interruptibility, the highest disclosure level provides a percentage accuracy of busyness, while the lowest level provides a simple abstraction (e.g. <33% is interpreted as the "user may not be busy"). We provide users a buffer for interpreting busyness by phrasing the disclosed information in terms of possibilities ("may not be busy") rather than absolute terms ("is not busy"). For location, the highest disclosure level uses the user's self-specified location tags while the lowest level indicates if the user is on or off campus. For active window, the highest disclosure level reports the name of the window in focus (e.g. "Mozilla Firefox – YouTube.com"), while the lowest level only reports the name of the application in focus (e.g. "firefox.exe").

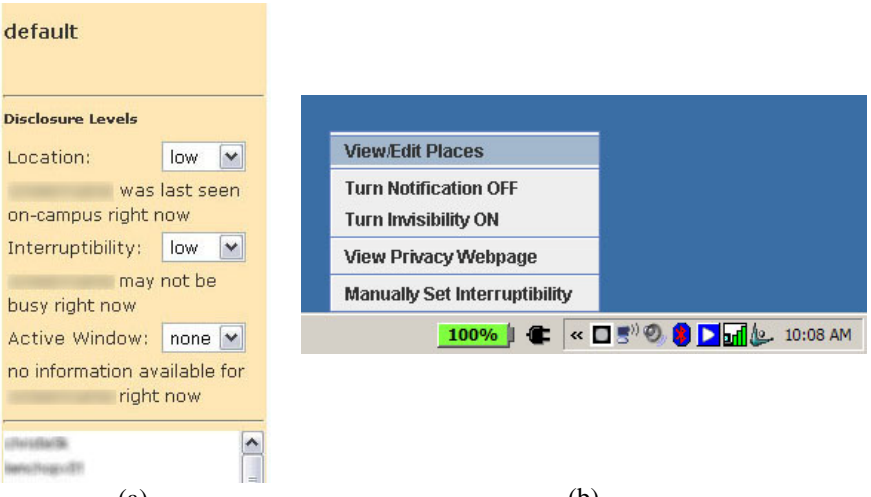
Table 1. Example of the different information abstractions based on the level of disclosure

Type	Level	Sample disclosure
Interruptibility	none	no information available for screenname
	low	screenname is somewhat busy 10 mins ago
	high	screenname is 60% available 10 mins ago
Location	none	no information available for screenname
	low	screenname last seen off-campus 10 mins ago
	high	screenname last seen at home 10 mins ago
Active Window	none	no information available for screenname
	low	screenname last used firefox.exe 10 mins ago
	high	screenname focused on Blackboard Academic Suite - Mozilla Firefox 10 mins ago

Groups-Based Privacy Policy Controls. We adapted a group-based approach based on prior work by Patil and Lai [9]. Users can specify privacy settings at any time via a web browser (see Figure 1a). Initially, a user's IM buddies are put in a 'default' privacy group, which uses the minimum information disclosure levels for all three information types. Users can create new privacy groups and populate them by moving buddies from the default group to any of their other newly created groups. If an unknown AIM user (a screenname who is not on the user's buddylist) requests information from *IMBuddy*, then he will automatically be added to the default group so that users can also adjust settings for strangers.

Through formative user tests, we found that people preferred using a vertically-oriented view for listing a group's privacy information, mostly because of its similarity to existing IM buddylist views. Within each group's container, drop-down controls let users set the disclosure level for each information type. As users change the disclosure level, dynamic "privacy transparency" feedback shows how their changes would affect the information disclosed to AIM buddies in that group.

Convenient Controls. While running the *IMBuddy* service, users have easy access to the privacy controls via a context menu (see Figure 1b). From this menu, users can: 1) suppress immediate notifications (as described in the next section), 2) turning on invisibility to prevent disclosing information (allowing for coarse-grained on/off control as suggested by [19]), and 3) quickly access their group privacy settings.



(a) (b)

Fig. 1. (a) Group-oriented view with group name, disclosure levels, and buddies for privacy control; (b) System tray icon allowing coarse-grained control and access to privacy settings

3.2 Feedback

IMBuddy supports three types of feedback: 1) informing users what information was disclosed to the requestor (disclosure history), 2) informing users when their information is being disclosed (notification), and 3) facilitating conversational grounding by informing users what others know about them (social translucency).

Disclosure History. The disclosure history is part of the privacy settings webpage, and provides a quick view of who has requested a user's information and what was disclosed (see Figure 2). From formative user tests, we found that people preferred viewing their disclosure history by date and buddy name as opposed to by information type or group. Our participants also rated the need to quickly view anomalies (based on the number of information requests) as an important privacy feedback feature. Moreover, our users found the relative amount of queries was more interesting than the absolute number. To visualize this, an at-a-glance feature using color highlights to indicate the number of requests was preferred over using the number of requests, with one participant saying that it "makes it easy to see who the stalkers are." As such, we see the disclosure history as an important feature for users to gauge if there are any problems in their privacy control settings. We note that our design used static thresholds for color highlighting, but more dynamic coloring schemes could be used.

Notifications. When someone requests a user's information, a bubble popup notification provides real-time feedback showing what was disclosed (see Figure 3a). These notifications remain on-screen if the user is not interacting with the computer (e.g. they are away from their computer at the time of disclosure). By not having an automatic notification dismissal, users have a chance to notice that a disclosure occurred while they were away and can readjust their privacy settings, if needed.

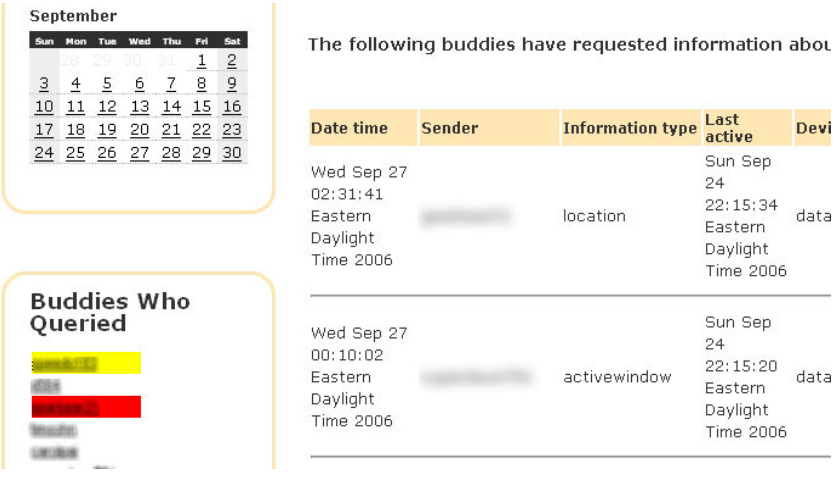
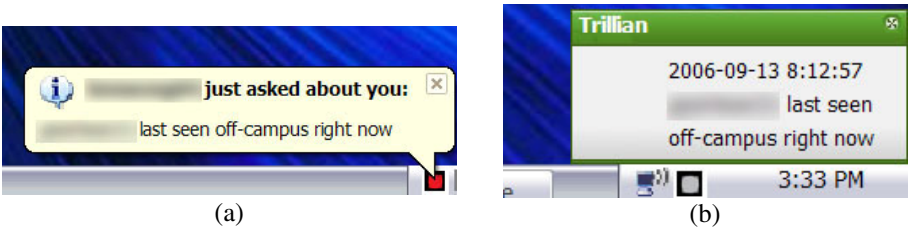


Fig. 2. The Disclosure History Page lets users see who has seen what, and when

We have also incorporated a non-distracting peripheral notification. When a disclosure occurs, our system tray icon changes from a white dot to a red dot, mimicking the red light used to indicate active recording status in recording equipments. This icon change alerts the user that their information is being recorded, accessed, and can be potentially sent to their buddies (depending on their privacy control settings). Moreover, this peripheral notification becomes the primary notification mechanism for users, if they choose to turn off bubble notifications.



(a) (b)

Fig. 3. (a) Bubble notifications provide immediate feedback on requests; (b) A popup is also displayed when a conversation occurs after a buddy has made an inquiry

Social Translucency. We also provide a notification reminding users what their buddies know about them when a conversation starts (see Figure 3b). This feedback mechanism provides conversational grounding [20] as well as social translucency of what information buddies have been requesting [3]. Using this information, users are less likely to be confused by their buddies' understanding (or lack thereof) of their current communication context. Furthermore, if people wish to provide a white lie while chatting, they will know the boundaries of which they can plausibly lie. For example, a person would not lie and say they were on campus if the notification said that the other person saw that they were at home. During our field deployments, we

provided these IM-based notifications by having our participants install a plugin we developed for Trillian Pro, a commercial IM client [21].

4 System Implementation

The *IMBuddy* system consists of three parts: an *IMBuddy* AIM Bot ("*imbuddy411*"), an *IMBuddy* server, and an *IMBuddy* client running on each participant's machine.

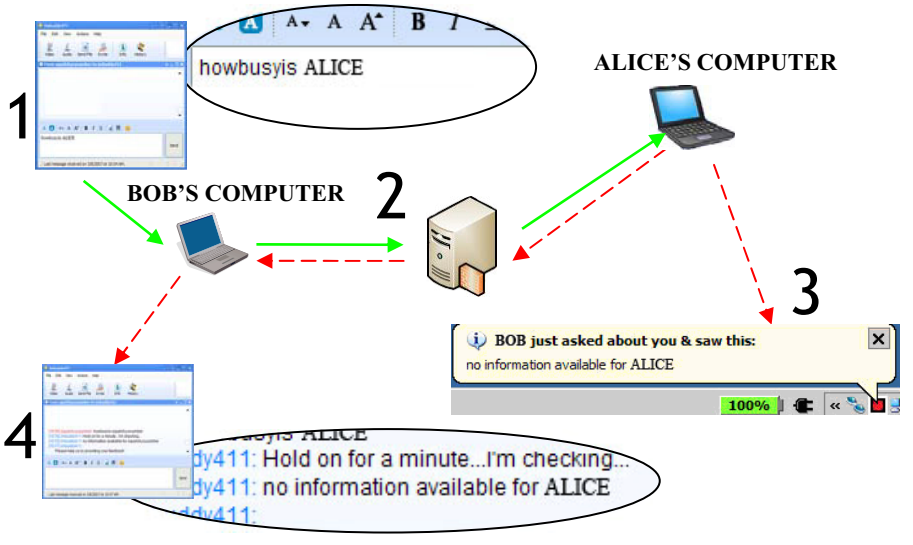


Fig. 4. (1) Bob queries on the busyness of Alice by typing "howbusyis ALICE" to *imbuddy411*; (2) *imbuddy41* passes the request to the *IMBuddy* server, which forwards it to the appropriate *IMBuddy* client to process the request; (3) Alice's client responds to the request and alerts Alice of the information that is being disclosed to Bob; and (4) *imbuddy411* then displays the privacy-filtered response from the client or database to Bob's chat window

Any AIM user can request a user's information by typing a command in a chat window to *imbuddy411* (implemented using JAIMBot, an open-source Java-based AIM library [22]) (see Figure 4). For example, he can type "*howbusyis* X" to get X's interruptibility, where X is the screenname (step 1). *imbuddy411* passes this request to the *IMBuddy* server, which then communicates with the appropriate *IMBuddy* client to retrieve the appropriate context information (step 2). The *IMBuddy* client notifies the user of the disclosure and relays the information back to the *IMBuddy* server (step 3). Based on the user's privacy settings, the *IMBuddy* server reports the privacy-filtered response back to the requester in the original chat window (via *imbuddy411*). Information requests are also stored in a MySQL database on the *IMBuddy* server, which lets the server share the most recent disclosure information if a user is offline.

The *IMBuddy* client software runs as a background process that collects interruptibility, location, and active window information. We use Subtle [23], a toolkit which uses sensor-based statistical models, to collect active window data and to

estimate a user's interruptibility. When tested with a group of 10 office workers, Subtle is capable of reaching 80% accuracy in predicting interruptibility. The model we used is built with data from human resource personnel and two graduate students [24]. Location estimates are done using a two-pronged approach. Because our participants are college students, the first level of location abstraction checks to see if users are on or off campus by determining if their IP address is within the university's subnet. To provide more precise location information, we rely on Place Lab [25] to sense nearby wireless access points. When our software sees a new set of wireless access points, we prompt users to provide a location tag. Later, we use Place Lab to recognize when the user returns to that location, so that we no longer need to prompt the user again. The *IMBuddy* client is also responsible for providing notifications, along with locally storing data to provide social translucency for IM conversations.

The *IMBuddy* server hosts the privacy control and history disclosure webpage and is implemented using Ruby on Rails and a MySQL database.

5 First Deployment and Redesign

To evaluate our feedback and control mechanisms, we recruited ten undergraduate students to use the *IMBuddy* system for a period of two weeks. We specifically chose undergraduates who were active AIM users that used IM for both social and work related purposes. On average, these participants are medium to heavy IM users; they have 90 buddies and 1300 incoming/outgoing messages a week. Based on the Westin Privacy Survey, these participants all fall in the Pragmatic category.

On the first day of the study, we installed the *IMBuddy* client software on each participant's laptop. They were also asked to set up their initial privacy groups by moving their buddies from the default group into any newly created groups and/or changing the settings for the default group. Participants were told that, throughout the study, they can change their settings by creating/deleting groups and moving buddies around anyway they like. For the purposes of our study, we wanted to have an initial set up so we could see how the initial groups change over the course of the study.

To introduce our *IMBuddy* service to our participants' buddies, we included a short description about the service in each participant's IM profile. However, because our participants said their buddies do not often check profiles, we modified our Trillian plugin to also advertise the *IMBuddy* service whenever an IM conversation is started.

There were a total of 242 individual queries made to *IMBuddy*. The breakdown of the different information types that were requested include: 66 for interruptibility, 104 for location, and 72 for active window. Since information requesters can ask for multiple types of information (for a given subject) in one session, we grouped such queries as a single instance. In all, there were 117 instances of use and on average two types of information were queried per use. 43 of those instances were times when *IMBuddy* disclosed information stored in the database (i.e. when users were not online or running our client). There were 53 distinct screen names who queried *IMBuddy* and 13 of those were repeat users.

A total of 43 groups were created. On average, there were 4.3 groups ($\sigma = 2.5$) per participant. One participant had only one group (default) and said that besides his active window, he was fine with anyone seeing his information. Other participants

had group names that contain keywords relating to class, major, clubs, gender, work, location, ethnicity, and blood relations. 6 of the 43 groups disclosed no information, while 7 of them disclosed all three information types at the highest level.

5.1 Findings

During a mid-deployment interview, we reviewed the disclosure history with each of the participants. At the end of the study, each participant completed a Likert-style questionnaire, where they were asked to rate 15 statements (where 1=strongly disagree and 5=strongly agree).

Our participants agreed that the three information types being disclosed were all potentially sensitive information that they would not carelessly disclose (interruptibility: 3.6/σ=1.3, location: 4.1/σ=1.1, active window: 4.9/σ=0.3). However, despite the potential sensitivity of this information, our participants said they were comfortable with their privacy settings for *IMBuddy* (4.1/ σ=0.9).

We found that our group-based control was intuitive to our users because they were used to similar levels of control from other sites and applications (*e.g* LiveJournal). They agreed that our privacy controls were easy to understand (4.4/σ=0.5) and easy to modify (4.2/σ=1.0). Users did, however, express a desire to be able to self-set interruptibility level, in the same way that they could self-tag location.

In terms of feedback, most users felt that they had a good sense of who had seen their information (3.9/ σ=1.2), and all of them had reviewed their disclosure history at least twice in the two weeks. They reviewed their disclosure log usually after noticing a query, which prompted them to find out what other information was disclosed since they last checked the disclosure history. The participants who gave low scores for this question indicated a need for a fourth type of feedback (that we later implemented), informing them about disclosures that occurred while their computers were off.

Our users did not feel that the notifications were problematic. For example, one user said, "[the notification was] at a good spot to ignore it if I wanted to." One participant did express concerns if the frequency of use increased: "if it were to happen all the time, then it might get annoying." One solution for this is to summarize disclosure histories. One participant suggested that "it would be cool if it was like summarized, like your location has been checked like 5 times, like something like that. I wouldn't want like it all to be listed. It would be too much." In specific cases where malicious users query the AIMBot and bombard users with unwanted notifications, one solution could be to have a blacklist where no information is disclosed and no notification is shown for blacklisted users; this is similar to the blocking option that current IM clients already have.

5.2 Redesign

The survey results indicated that users were mostly satisfied with the privacy controls. They felt the controls were easy to use and understand. Most importantly, they were comfortable with their privacy control settings. Therefore, in our redesign, we only added minor changes to the control mechanisms, such as allowing users to correct the interruptibility information being disclosed. Instead, our redesign focused on the reported need for different types of privacy feedback mechanisms.

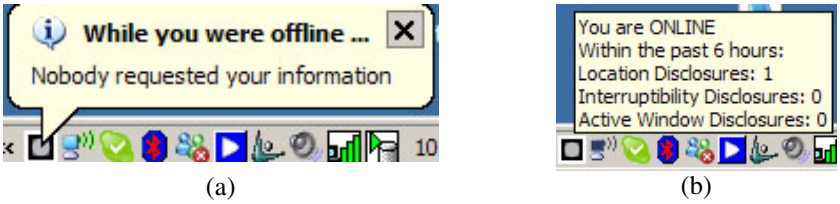
System and Control Modifications. One concern with our first deployment was the inability to correct the information being disclosed. While our system would ideally only disclose accurate contextual information, we found that interruptibility was often not accurate enough for our users. Hence, participants from our first deployment requested the ability to self-tag their interruptibility, much in the same way that we allow for location self-tagging. Thus, we modified our *IMBuddy* client to allow users to manually set their interruptibility though the client's context menu. All manual interruptibility settings would only last for one hour, after which the user's interruptibility would revert back to the system-inferred value.

We also found that the highest level of interruptibility disclosure (a percentage) is generally not as useful as an abstract text description of the user's interruptibility (e.g. "not busy"). To address this, we modified interruptibility's highest disclosure level to include both the percentage and a brief text description.

Lastly, we modified the client to auto-update the user's contextual information to the server every five minutes, as opposed to only when a query is sent. This way, the latest information in the server will remain reasonably up-to-date, so that information requestors can still get useful information when the user's computer is offline.

Additional Feedback. We added two new feedback mechanisms. The first provides feedback to the users when they logon to the system, showing them the number of information requests that occurred while they were offline in a bubble notification (see Figure 5a). The purpose for this feedback is to provide the users a better sense of how their privacy was handled while they were offline.

The second feedback mechanism appears when a user mouses over the client's system tray icon (see Figure 5b). A small tool-tip popup window appears, showing the number of requests for each information type within the past 6 hours. We designed this mechanism to provide a lightweight summary of disclosure history. This is especially useful if the users have not been actively keeping track of the disclosures (e.g. because they were away from their computer for an extended period of time).



(a) (b)

Fig. 5. (a) Feedback after logging-on; (b) a mouse-over notification providing a summary

6 Second Deployment

To evaluate our modified redesign, we conducted a second field study and deployed our contextual IM system to 15 students for four weeks. These are different participants than the ones from our first deployment. These participants, on average, are medium to heavy IM users; they have 120 buddies and 1580 incoming/outgoing

messages a week. These users were also all Pragmatic according to the Westin scale. We used the same advertising techniques as in the first deployment.

In all, there were 140 instances of use. 74 of those instances disclosed information stored in the database (i.e. when users were not online or running our client). The breakdown of the requested information types were: 67 for interruptibility, 175 for location, and 79 for active window (for a total of 321 individual queries). There were 61 distinct screennames who queried *IMBuddy*. As expected, some users queried the system due to novelty, but there were 15 repeat users who continually used the system throughout the duration of the study.

A total of 56 groups were created. On average, 3.3 groups ($\sigma = 1.3$) were created per user. Groups were again separated by an array of factors: class, major, clubs, gender, location, ethnicity, and blood relations. When asked, users often described a sense of closeness as the underlying separating factor between the groups. 6 of the 56 groups disclosed no information, while 10 of them set all three information types at the highest disclosure level. All but 2 default groups disclosed no information, while others allowed for at least a medium disclosure level for interruptibility.

For this deployment, we again focused our evaluation on the privacy control and feedback mechanisms provided in *IMBuddy*. Our metrics of evaluation include awareness (how successful our system was in keeping our users informed), convenience and ease of use (how easy it was to understand and use the controls), and appropriateness (how the users felt about the disclosed information).

We solicited participants' thoughts using multiple evaluation techniques. We conducted interviews and used surveys/questionnaires[1]. We also created a "stalker-bot" (*jasonkats722)*, to test the effectiveness of our feedback and notification systems and to observe how our participants would react to an unknown and potentially malicious user. The stalker-bot was implemented as another AIMbot which would query *IMBuddy* for different users and about different information types at random times. The stalker-bot was deployed near the end of the study, when our participants were already familiar with how *IMBuddy* works and had enough time to settle into a "comfortable" privacy setting. On average, the stalker-bot made 2 sets of queries a day, asking for two or three types of information per session.

6.1 Evaluation of Control

To evaluate our control mechanism, we asked our participants to comment on three things: 1) the general usability of the controls (e.g. do users know how to modify the settings and is the design easy to use), 2) their comfort level in regards to the stalker-bot, and 3) their perception of the information disclosed.

Usability. Similar to our first study's results, the participants in our second deployment again felt the privacy controls were intuitive, easy to use, and allowed for easy and quick corrections to any errors. The extended length of the second study combined with the increased number of participants, strengthened our findings from the first-iteration. Our survey questions regarding the understandability of our privacy controls and the ease of changing privacy policies are both highly rated ($4.5/\sigma=0.7$).

[1] The means reported here use the same 5-point Likert scale as in our first field deployment, unless otherwise noted.

During our interviews, participants repeatedly made statements such as "I really liked the privacy settings the way they are. I thought they were easy to use, especially changing between privacy settings." Ability to access the control easily was also mentioned: "I felt pretty comfortable with using it because you can just easily modify the privacy settings." Another participant concurs, saying: "it's flexible; you can create as many groups as you want. Moving people around is relatively easy. Since it's on a website, it's not like you have to open up an application."

However, a couple of participants did comment that setting up their initial privacy groups was a bit tedious. "It's time consuming, if you have a long buddylist, to set up for each person." Such comments suggest a need to reduce initial costs that may occur with using group-level control. There were also desires by certain participants to allow for more levels of disclosure granularity. Specifically, a few participants wanted one more disclosure levels for location information, where users could say that they were around a certain place (versus at a specific place).

Comfort. Our participants said they were comfortable with their privacy settings for *IMBuddy* ($4/\sigma=0.9$). Moreover, users' comfort levels were not changed after introducing the stalker-bot. Participants who did not notice the stalker-bot, reacted no differently than hearing about any other user querying for their information. They reasoned that *jasonkats722* was perhaps one of their buddies, or that he was an old friend that was no longer on their buddy list. Most important is that they were not concerned. They were confident in their privacy control settings and it did not matter to them that a potential stranger had been checking their information multiple times: "I know they won't get any information, because I set the default so they won't be able to see anything."

Appropriateness of Disclosures. For our mid-deployment interview, we asked our participants to describe scenarios where they felt that: 1) the information disclosed was inappropriate (either too much or not enough information was provided), and 2) the information disclosed was just right and/or extremely useful.

Overall, the participants were not able to state any particular incidences where they felt the information disclosed was inappropriate. This is partly because the overall number of queries was not that high, but it also reflects that users felt comfortable with the information their buddies would potentially see. One user mentioned that he experimented with the system and realized that active-window queries could lead to potentially embarrassing information disclosures (e.g. someone could find out if he happened to be visiting a porn site). While he had initially allowed his friends to see the most detailed information regarding his active window, after this discovery, he went back and changed the settings to prevent potential embarrassing incidents from occurring. Another participant discussed how she lowered her privacy settings for a particular classmate who frequently asked for her information because she felt he did not need such detailed information as she was originally disclosing.

While the amount of use has not been extremely high, we were still able to witness incidents where participants found contextual IM to be very useful. We describe one such scenario below, where the participant's buddy used the service to coordinate with our participant, without bothering them directly.

Quote 3 <participant L> *"Someone asked where I was [using IMBuddy], did not IM me and then showed up there... there is a room that I hang out in a lot and she comes there a lot. But you need a key to get in, and I have a key but she doesn't. She's not going to show up if there is no one there that has the key. So she'll check if I'm there and then come…and I knew [that she had asked for my information] because it shows me in that little thing [notification bubble]…she would complain when it is not accurate and stuff, like I'll leave my computer on with my IM up and go and get food or something, and she'll be in the room when I get back, and she'll be like 'it told me you're here and you're not.'"*

As indicated by the quote above, one complaint was actually the inaccuracy of some information disclosures as opposed to its inappropriateness. Inaccuracies of information existed in two forms: the system-inferred interruptibility is not always 100% accurate, and the location accuracy is limited by the extent to which the user takes their laptop with them.

According to our participants, the most useful information type is location. Location was preferred over availability in terms of utility because most IM users are accustomed to sending an IM message (e.g. "are you free") to determine availability, which is an interruption in of itself. One participant said "I don't really get the point for how busy I was, because people would IM me regardless." Location is also more useful than active window because our participants did not use IM often in group-work scenarios, where awareness of each other's task might be more helpful.

During the study, *IMBuddy* would also randomly survey information requestors to get a sense of the appropriateness of the disclosed information using a 5-point Likert scale, where 1 is "wanted more information", 3 is "obtained just the right amount of information", and 5 is "got a lot more than asked". Based on 61 logged entries, the mean was 2.47 ($\sigma = 0.91$). Since we did not specifically indicate to these people the range of responses they could have gotten, one might question if most have selected the 3 simply because of a lack of comparison point. However, the average rating does suggest the right level of information was disclosed.

6.2 Evaluation of Feedback

We evaluated the awareness of disclosure both in terms of user feedback (using our survey results), and by our users' reactions to our stalker-bot, *jasonkats722*.

General Awareness of Disclosures. From our first field deployment, users reported having a fairly good sense of who had seen what ($3.9/\sigma=1.2$). It was, however, apparent from our interviews that some participants desired different types of feedback from what we had designed. Mainly, participants desired feedback to support their awareness of disclosures when they are not able to monitor the disclosure bubble notifications. With the two newly added feedback mechanism, the mean agreement rating to the question "while using the system, I always have a good sense of who has seen what" increased to 4.1 ($\sigma=0.8$).

We speculated that if asked, participants would claim that they had found all of the feedback mechanisms to be helpful. Therefore, to gain a better understanding for

which feedback was more essential, we asked our participants to rank the 6 different types of feedback mechanisms that we had designed (with 1 being most useful). The average rankings from most useful to least useful were: bubble notification, 1.6 (σ=0.6); disclosure log, 1.8 (σ=1.3); mouse-over notification, 3.7 (σ=1.0); offline statistic notification, 4 (σ=1.4); social translucency Trillian tooltip popup, 4.8 (σ=1.1); and peripheral red-dot notification, 5.4 (σ=0.7).

Awareness of Stalker-Bot. One of the main purposes for using awareness as a metric for evaluating privacy control and feedback is to ensure the users are able to detect if there are any cases of misuse. By doing so, users can take the necessary actions to protect themselves in a timely fashion. We tested user awareness by deploying a stalker-bot named *jasonkats722* 2-5 days before the end of the study[2]. During the post-study interview, we asked our participants to describe their relationship with a list of screennames who had previously queried for their information. As we proceeded down the list, we focused on *jasonkats722* and asked follow-up questions to better understand how our participants' reaction to his stalker behavior.

There were 12 participants who noticed *jasonkats722* (1 participant was out of town and did not use the system during that period). Of these, only a couple of them did not think too much about it, since they only noticed 1 or 2 queries made by *jasonkats722* and assumed it was some random person or another participant's buddy testing out the system: "It does bother me that someone I don't know has looked at it, but the fact that I've gone in and set my settings appropriately, minimizes that." Other participants, however, did go back to the disclosure log in an attempt to figure out what *jasonkats722*'s motivation may have been. One user even went as far as attempting to message *jasonkats722* whenever he went online.

Quote 1 <participant A> *"I think yesterday was the first time that I'd noticed him and I think yesterday was the first time that happened. I then went to my privacy settings to check, cuz I'd forgot what his screenname was. I went and checked his screenname. Added to my buddylist and asked who he was but I never got a reply. He would sign on and off…it was the popup bubble [that first notified me]…first time I thought it was unusual, but I didn't do anything. But then I saw it the second time like 10 minutes later, so I was intrigued, wanted to know why this person who I don't know is asking about me."*

We asked our users about the potential use of a blacklist, an idea that we got from our first iteration, where a particular screenname would not get any information and participants would also not be bothered by disclosure requests from that screenname. While participants liked the idea of screening certain users from accessing any information, they still wanted to know who was asking for their information.

Quote 2 <participant A> *I wouldn't like in real life if someone randomly asked where I was, but I would like to know who these people are. Like my friends would tell me someone was asking about me, and tell me who that person was. But over the internet I can't do that, so I have to find out myself."*

[2] Some participants ended slightly earlier than others.

7 Discussion

The goal of this work is to provide a better understanding about the types of control and feedback mechanisms that would be valuable and necessary for privacy-sensitive contextual information disclosure through instant messaging. While it is unfortunate that *IMBuddy*'s use was not as high as we had hoped for, we were still able to draw informative findings using our qualitative data collected from people's perception of our control and feedback mechanisms. Two groups of student users interacted with our service and design for a period of 2-4 weeks, in everyday social and work settings, and were exposed to potential misuse by strangers. Our first iteration indicated the need for more feedback to provide disclosure awareness while the user is away or offline. Our second iteration explored use of controls and feedback in more depth, through more users and longer use.

7.1 Controls

Users from both deployments thought our controls were easy to understand and use. They were able to disclose their information at a level they were comfortable with, while still getting value from using the system. Even though we cannot make any strong claims stating that our control mechanisms offered the best balance between usefulness and appropriateness and will generalize to more complicated information types, we do believe from our deployments that it provides a set of baseline mechanisms for future work to be compared against. The coarse-grained invisible control was useful for providing users some "alone time" and the notification-off control was useful for preventing distractions.

Although no previous research has clearly demonstrated that group-based privacy control is sufficient for contextual information disclosure, our work does providing promising evidence that it works well in practice. One of the primary reasons is that it is easy to understand. People have been using groups to organize IM buddylists, as well as other social application (e.g. flickr and LiveJournal).

One key issue about using group-based privacy controls is how to decrease the initial set-up costs. Given that our participants had on average 90+ buddies, creating groups and placing their buddies into groups took some time. One idea is to bootstrap the system using existing IM buddy groups and screennames. However, from our deployments, we found that IM buddy groups are quite different from the privacy groups created in *IMBuddy*. Groups created in *IMBuddy* tend to be separated by levels of closeness. On the other hand, IM buddy groups are typically separated by where and how the user knows the buddy. This distinction prevents users from leveraging their current IM buddy groups to simplify the process of creating their privacy groups.

Our evaluation also indicates that when preloading the system with an initial group of buddies, those preloaded buddies should be automatically placed in a group separate from the "default" group. This would differentiate between actual strangers and buddies. In addition, based on our interviews, there is evidence that suggests the need for a blacklisted group. Disclosure requesters from that group could potentially receive false information both to maintain plausible deniability and to prevent requesters from realizing they are in the blacklisted group. Such a design would fulfill

the recommended design guideline of supporting deception [11]. But such mechanisms need to be carefully designed.

Another concept worth exploring is allowing buddies to have multiple memberships. That was suggested by participants in both deployments. It makes sense why this particular use of groups would be intuitive. We can have more than one type of relationship with a buddy. Depending on the context, we might want to give certain groups that a buddy belongs to more control than others. Thus, by allowing for multiple memberships, we can increase the flexibility of group-based privacy controls. However, we would also need to then address how to resolve potential privacy policy conflicts. Nevertheless, this idea deserves further exploration as it has not been explore in prior work on group-based privacy configuration designs.

7.2 Feedback Mechanisms

In both deployments, our surveys indicated that participants thought they had a good sense of who had seen their information. In our first iteration, there were four types of feedback: disclosure history, bubble notification, peripheral notification (the red dot) and Trillian tooltip popup supporting social translucency. Our first deployment led to two additional feedback designs, an offline statistics notification and a mouse-over notification. While the second deployment's survey response to the same question was slightly higher, it was not statistically significant. The rankings of the 6 different types of feedback indicated that the bubble notification, as expected, is the most important notification for our users. It allows for immediate feedback regarding who has seen what, giving users an opportunity to react to the disclosure if necessary. This suggests that future contextual IM services should minimally include this type of feedback mechanism for their users.

Our exploration with the stalker-bot *jasonkats722* suggests a good start for modeling when to alert users regarding potential misuse, namely to provide alerts based on if the information queries has occurred more than once and how much time has elapsed since the last query. Queries by strangers should also result in more immediate alarms than by someone who is on the user's buddylist. One participant mentioned this potential design of stalker alert:

Quote 3 <participant M> *I think it would be good like if a strangers asks and if they don't find anything, or that you would ignore it, maybe there's some kind of threshold so if they keep asking, like I don't know how many times…the same guy keeps asking, and I don't know him then it would let me know like hey there's this guy, you might want to check into this see if someone you know is trying to get a hold of you or if it's someone you don't know that is trying to stalk you.*

8 Conclusions

In this work, we present the design of privacy controls and feedback mechanisms using a contextual IM service called *IMBuddy*. We conducted an initial two week field study of our systems and re-iterated our system design based on our initial findings. We then deployed our system in a second field study, lasting 4-weeks with 15 users. Our findings suggest that *IMBuddy* successfully provided effective

awareness for our users (e.g. participants were aware of when and to whom their information was disclosed to) in addition to intuitive, easy-to-use privacy controls that enabled them to configure their privacy settings to a comfortable level. Furthermore, *IMBuddy* provides positive evidence that group-based privacy configuration is intuitive and sufficient for our contextual IM framework. We believe results from this study can and should be extended to future designs of contextual IM and contextual telephony applications.

9 Future Work

We plan to explore how to encourage more *IMBuddy* use to further validate our findings. It is not clear if the problem lies with a lack of critical mass, or if using an AIM Bot to disclose contextual information is an inappropriate design metaphor. Greater use will also facilitate longer and larger field trials that will help us more fully understand the intricacies of privacy, privacy controls, and social perceptions.

Acknowledgments. This work is funded in part by NSF grants CNS-0627513, IIS-0534406, and ITR-032535. We also thank the contributors to Place Lab, JAIMBot, and the jdic project, as well as James Fogarty, Joe Tullio, Ian Li, Scott Hudson, Robert Kraut, and our Common Meeting group members.

References

1. Isaacs, E., Walendowski, A., Whittaker, S., Schiano, D.J., Kamm, C.: The character, functions, and styles of instant messaging in the workplace. In: ACM conference on Computer Supported Cooperative Work (CSCW), pp. 11–20. ACM Press, New York (2002)
2. Nardi, B., Whittaker, S., Bradner, E.: Interaction and Outeraction: Instant Messaging in Action. In: ACM Conference on Computer Supported Cooperative Work (CSCW), pp. 79–88. ACM Press, New York (2000)
3. Erickson, T., Kellogg, W.A.: Social translucence: an approach to designing systems that support social processes. In: TOCHI, vol. 7, pp. 59–83 (2000)
4. Isaacs, E., Walendowski, A., Ranganathan, D.: Hubbub: A Sound-Enhanced Mobile Instant Messenger that Supports Awareness and Opportunistic Interactions. In: ACM Conference on Human Factors in Computing Systems (CHI), pp. 179–186. ACM Press, New York (2002)
5. Scupelli, P., Kiesler, S., Fussell, S.R., Chen, C.: Project view IM: a tool for juggling multiple projects and teams. In: Extended Abstracts of ACM Conference on Human Factors in Computing Systems (CHI), ACM Press, New York (2005)
6. Tang, J.C., Yankelovich, N., Begole, J., Kleek, M.V., Li, F., Bhalodia, J.: ConNexus to awarenex: extending awareness to mobile users. In: ACM Conference on Human Factors in Computing Systems (CHI), pp. 221–228. ACM Press, New York (2001)
7. Bellotti, V., Sellen, A.: Design for Privacy in Ubiquitous Computing Environments. In: Third European Conference on Computer Supported Cooperative Work (ECSCW), pp. 77–92 (1993)
8. Gaver, W., Moran, T., MacLean, A., Lovstrand, L., Dourish, P., Carter, K., Buxton, W.: Realizing a video environment: EuroPARC's RAVE system. In: ACM Conference on Human Factors in Computing Systems (CHI), pp. 27–35. ACM Press, New York (1992)

9. Patil, S., Lai, J.: Who gets to know what when: configuring privacy permissions in an awareness application. In: ACM Conference on Human Factors in Computing Systems (CHI), pp. 101–110. ACM Press, New York (2005)
10. Bly, S.A., Harrison, S.R., Irwin, S.: Media spaces: bringing people together in a video, audio, and computing environment. ACM Communications 36, 28–46 (1993)
11. Iachello, G., Smith, I., Consolvo, S., Chen, M., Abowd, G.D.: Developing privacy guidelines for social location disclosure applications and services. In: Symposium on Usable Privacy and Security (SOUPS), pp. 65–76. ACM Press, New York (2005)
12. Lederer, S., Hong, J.I., Dey, A., Landay, J.A.: Personal Privacy through Understanding and Action: Five Pitfalls for Designers. Personal and Ubiquitous Computing 8, 440–454 (2004)
13. Begole, J.B., Tang, J.C., Smith, R.B., Yankelovich, N.: Work rhythms: analyzing visualizations of awareness histories of distributed groups. In: ACM Conference on Computer Supported Cooperative Work (CSCW), pp. 334–343. ACM Press, New York (2002)
14. Terveen, L., Akolkar, R., Ludford, P., Zhou, C., Murphy, J., Konstan, J., Riedl, J.: Location-Aware Community Applications: Privacy Issues and User Interfaces. In: Location-Privacy Workshop (2004)
15. Hong, J.I., Landay, J.A.: An Architecture for Privacy-Sensitive Ubiquitous Computing. In: Second International Conference on Mobile Systems, Applications, & Services (MobiSys), pp. 177–189 (2004)
16. Palen, L., Dourish, P.: Unpacking "Privacy" for a Networked World. In: ACM Conference on Human Factors in Computing Systems (CHI), pp. 129–136. ACM Press, New York (2003)
17. Langheinrich, M.: Privacy by Design: Principles of Privacy-Aware Ubiquitous Systems. In: Abowd, G.D., Brumitt, B., Shafer, S. (eds.) Ubicomp 2001. LNCS, vol. 2201, pp. 273–291. Springer, Heidelberg (2001)
18. Nguyen, D.H., Mynatt, E.D.: Privacy Mirrors: Making Ubicomp Visible. In: ACM Conference on Human Factors in Computing Systems (CHI), Workshop on Building the User Experience in Ubiquitous Computing (2001)
19. Lederer, S., Hong, J.I., Dey, A.K., Landay, J.A.: Personal privacy through understanding and action: five pitfalls for designers. Personal and Ubiquitous Computing 8, 440–454 (2004)
20. Clark, H.H., Brennan, S.E.: Grounding in communication. In: Resnick, L., Levine, J., Teasley, S. (eds.) Perspectives on Socially Shared Cognition, pp. 127–149. American Psychological Society, Washington, DC (1991)
21. Cerulean Studios - Trillian Pro, http://www.trillian.cc
22. Java AIMBot, http://sourceforge.net/projects/jaimbot
23. Fogarty, J., Hudson, S.E.: Toolkit support for developing and deploying sensor-based statistical models of human situations. In: ACM Conference on Human Factors in Computing Systems (CHI), pp. 135–144. ACM Press, New York (2007)
24. Tullio, J., Dey, A.K., Chalecki, J., Fogarty, J.: How it works: a field study of non-technical users interacting with an intelligent system. In: ACM Conference on Human Factors in Computing Systems (CHI), pp. 31–40. ACM Press, New York (2007)
25. LaMarca, A., Chawathe, Y., Consolvo, S., Hightower, J., Smith, I.E., Scott, J., Sohn, T., Howard, J., Hughes, J., Potter, F., Tabert, J., Powledge, P., Borriello, G., Schilit, B.N.: Place Lab: Device Positioning Using Radio Beacons in the Wild. In: Third International Conference on Pervasive Computing (Pervasive), pp. 116–133 (2005)