I Share, You Care: Private Status Sharing and Sender-Controlled Notifications in Mobile Instant Messaging

HYUNSUNG CHO, KAIST, Republic of Korea
JINYOUNG OH, KAIST, Republic of Korea
JUHO KIM, KAIST, Republic of Korea
SUNG-JU LEE, KAIST, Republic of Korea

While mobile instant messaging (MIM) facilitates ubiquitous interpersonal communication, its constant connectivity could build the expectation of an immediate response to messages, and its notifications flood could cause interruptions at inopportune moments. We examine two design concepts for MIM—*private status sharing* and *sender-controlled notifications*—that aim to lower the pressure for an immediate reply and reduce unnecessary interruptions by untimely notifications. Private status sharing reactively reveals a customized status with a selected partner(s) only when the partner has sent a message. Sender-controlled notifications give senders the control of choosing whether to send a notification for their own messages. We built *MyButler*, an Android app prototype that instantiates these two concepts and integrated it with KakaoTalk, a commercial MIM app. During a two-week field study with 11 pairs (5 couples and 6 friend pairs), participants expressed themselves through a total of 210 different statuses, 64.3% of which indicated the current activity or task of the user. Participants reported that private status sharing enabled them to explain their unavailability and relieved the pressure and expectations for timely attendance. We reveal more findings on the types of privately shared statuses and their roles in MIM communication; the in-situ behaviors and patterns of using sender-controlled notifications; and the motivations of MIM users in choosing whether to alert their messages. In terms of message notifications, senders chose to send 25.4% of the messages without any notification. We found that senders' decisions to alert are affected by the receiver's status, their own status to chat, and the possibility of message content exposure to others through notifications. Based on our findings, we draw insights into how the concepts of private status sharing and sender-controlled notifications can be applied in future designs and explorations.

CCS Concepts: • **Human-centered computing → Empirical studies in HCI**; *Empirical studies in collaborative and social computing*; *Ubiquitous and mobile computing systems and tools.*

Additional Key Words and Phrases: Mobile Instant Messaging; Social Awareness; Notifications; Interruptions

ACM Reference Format:
Hyunsung Cho, Jinyoung Oh, Juho Kim, and Sung-Ju Lee. 2020. I Share, You Care: Private Status Sharing and Sender-Controlled Notifications in Mobile Instant Messaging. *Proc. ACM Hum.-Comput. Interact.* 4, CSCW1, Article 34 (May 2020), 25 pages. https://doi.org/10.1145/3392839

Authors' addresses: Hyunsung Cho, hyunsungcho@kaist.ac.kr, KAIST, Republic of Korea; Jinyoung Oh, jinyoungoh@kaist.ac.kr, KAIST, Republic of Korea; Juho Kim, juhokim@kaist.ac.kr, KAIST, Republic of Korea; Sung-Ju Lee, profsj@kaist.ac.kr, KAIST, Republic of Korea.

2573-0142/2020/5-ART34 $15.00
https://doi.org/10.1145/3392839

1 INTRODUCTION

Mobile instant messaging (MIM) is a popular interpersonal communication method today. Mobile instant messages and accompanying notifications are, however, known to "significantly disrupt performance on an attention-demanding task" [58] and thus could distract user's attention [33] and cause interruptions at inopportune moments [32, 45].

In remote communications where the sender and the receiver are not physically co-located, it is difficult, if not impossible, for senders to know whether the receiver would be interrupted at the moment. Besides, in current MIM systems, there is a lack of option for senders to control notifications to not disturb the receiver. Existing MIM apps (e.g., WhatsApp [26], Facebook Messenger [24]) provide availability indicators such as *read receipt* and *last active/seen time*. However, these approximations of availability do not always correctly reflect one's availability and are disclosed uniformly to any online friend. Prior work has discovered that these could yield the consequences of privacy concerns [50], feeling of being observed and patronized, and creating social pressure [48].

In order to increase social awareness between remote users, especially those in intimate relationships, previous research [2, 5, 16, 52] has explored how sharing diverse types of contextual information between users impacts the relationship in various ways. However, there is a lack of understanding on how sharing contextual information can alleviate the concerns of social pressure to immediately respond and interruptions related to MIM. Another branch of research has investigated context-aware notification systems to mitigate smartphone interruptions. These systems estimate the message receiver's current situation and the value of notification to defer [12, 21, 28, 41–43, 45, 47] or filter [13, 38] non-urgent notifications at inopportune moments. However, these could not analyze the *content* of a notification (e.g., message contents for MIM notifications) due to privacy issues [39] despite its important role in predicting receptivity to mobile notifications [13, 40].

We examine two design concepts to improve MIM communication by enhancing social awareness and reducing interruptions: (1) private status sharing and (2) sender-controlled notifications. The concept of private status sharing tackles the privacy concerns of existing systems that disclose the status continuously for any online friend. Private status sharing has three main characteristics: personal, custom, and reactive. A user configures a custom status to share personally with only allowed partner(s). Only when an allowed partner sends a message to the user, they see the configured status in reaction to the message. The concept of sender-controlled notifications aims to reduce unnecessary interruptions from untimely MIM notifications by including the message sender in the decision loop for the urgency of a message notification. Along with the status of the message receiver, the message sender is given two options, *Alert Now* and *Don't Alert*, to control the notification for their own message. The sender can decide whether to alert their message based on the given context of the receiver and the urgency of their message content.

We developed our Android app prototype *MyButler* that instantiates the conceptual ideas of private status sharing and sender-controlled notifications. MyButler augments the functions of the two design concepts on top of KakaoTalk, a dominantly popular commercial MIM app in Korea [9]. We evaluated MyButler on 11 pairs through a two-week field study of frequent KakaoTalk users. The user's custom status data and notification choice logs were analyzed along with the retrospective interview data to understand user behaviors and experiences with our system. We found that the privately shared statuses mainly served the roles of (i) relieving the burden and expectation of immediate response, (ii) advertising (un)availability, and (iii) sharing real-time status. For sender-side notification controls, when the participants made decisions between *Alert Now* and *Don't Alert*, they considered (i) whether the receiver's status indicates the notification might interrupt the receiver, (ii) whether it indicates the possibility of other people seeing their message, (iii) the importance or urgency of the message, and (iv) the sender's availability or willingness to chat.

Overall, MyButler provides a private, considerate, and intimate experience of status sharing and notification control in MIM. Our field study explores the in-situ user behaviors of using the new features. Our contributions are the following:

- We found that private status sharing was mostly used to share the current activity or task in intimate relationships, which often served as an explanation for one's unavailability. Sharing the reason for unavailability helped relieve the expectation and social pressure of immediate response in MIM communications.
- We revealed that a reactive way of sharing contextual information could be privacy-preserving, which might offer opportunities to mitigate the concern of feeling observed and surveilled often raised by automatic sharing of contextual information.
- We discovered opportunities for sender-controlled notifications in MIM that senders sometimes preferred to send their messages without notifications. Senders chose to not alert to avoid interrupting the receiver in inopportune moments, to hide messages from other people around, and when the senders were unavailable or unwilling to talk at the moment.

2 RELATED WORK

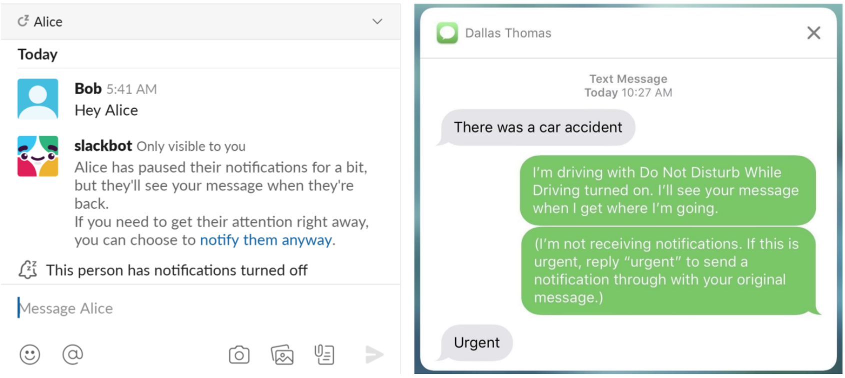
We discuss previous work on increasing awareness of availability in mobile instant messaging, supporting social awareness in remote communications and mitigating interruptions from smartphone notifications.

2.1 Increasing Awareness of Availability in MIM

Commercial MIM apps provide features that indicate one's availability. One example is *last active/seen time* in Facebook Messenger [24] and WhatsApp [26] that automatically tracks when a user has last opened the app. The *read receipts* is another popular availability indicator in Facebook Messenger, KakaoTalk, and WhatsApp. Each message is marked with whether the recipient has read the message and often used to infer the user's availability to chat. While serving as approximations of availability status, online activity information and read receipt accompany negative consequences such as stress [48, 54], privacy concern [8, 23, 48, 50, 54], frustration [8], feeling of being observed [48], patronized [48], and ignored [23]. Pielot et al. [48] introduced a machine learning-based prediction of attentiveness in MIM using smartphone use logs. It classifies a user's attentiveness into *high*/*low* classes and offers higher accuracy, less privacy invasion, and reduced social pressure over WhatsApp's *last seen* status. However, as this relies on a systematic prediction, there were concerns of "being afraid to create false expectations" when the prediction is inaccurate [48].

Status (e.g., KakaoTalk, Skype [57], Slack [25], Threads [27]) and *About* (e.g., WhatsApp) are also used to freely express oneself on the user profile. The status in the form of text or icon can be manually set by the user and shown on the user's profile for other users on the friends list. The current apps require a tedious sequence for status update such as first opening one's own profile, clicking the profile management button, selecting to edit the status, erasing the old and typing out the new status, and clicking the confirm button. Furthermore, they uniformly disclose the status to even acquaintances whom the user might not want to share with. This could discourage users from revealing their meaningful yet private status. These limitations of existing systems call for a better availability indicator that is more privacy-preserving and under the user's full control.

Instagram recently launched a new companion app Threads [27], a messaging app for "Close Friends" only. The members of the Close Friends list are manually selected by the user, and hence Threads enables a personal experience of messaging and sharing moments. In addition to the manual *Status* feature, Threads supports the *Auto Status* feature that automatically updates the



Bob 5:41 AM
Hey Alice

slackbot Only visible to you
Alice has paused their notifications for a bit, but they'll see your message when they're back.
If you need to get their attention right away, you can choose to notify them anyway.

This person has notifications turned off

Text Message
Today 10:27 AM

There was a car accident

I'm driving with Do Not Disturb While Driving turned on. I'll see your message when I get where I'm going.

(I'm not receiving notifications. If this is urgent, reply "urgent" to send a notification through with your original message.)

Urgent

Fig. 1. Screenshots of the Do Not Disturb (DND) mode in Slack (left) and the Do Not Disturb While Driving mode in Messages on iOS (right) [63]. When the DND mode is activated, Slack sends a slackbot message to the message sender that the user has paused notifications. When the DND While Driving mode is activated, Messages auto-replies the message sender to inform that the user is currently driving and can not receive notifications. The sender in both systems can bypass the mode and send the message notification by clicking the 'notify them anyway' button (Slack) or typing 'Urgent' (Messages).

user's status based on the accelerometer, location, and battery level information. It is yet unknown whether the privacy concerns related to automated availability indicators remain with *Auto Status*; how users practice status sharing among a close group of friends; and how the status impacts MIM communication.

Some messengers such as Slack, Skype, and Messages on iOS offer the Do Not Disturb (DND) mode. When the DND mode is activated, the apps indicate that the user has activated the DND mode through a DND icon on the profile (Slack and Skype) or an auto-reply message (Messages on iOS). Figure 1 shows Slack's DND mode on the left and Messages's DND While Driving mode on the right. In addition to updating the indicator shown to potential message senders, the DND mode also takes effect on the side of the message receiver by snoozing all message notifications. For example, when a Messsages user has turned on its DND While Driving mode, all Messages notifications are snoozed while driving and the system auto-replies a message sender informing that the user is currently driving and not receiving notifications. Messages as well as Slack further provides a way for a message sender to bypass the DND mode and send a notification for urgent matters.

Cho et al. [7] also introduced a design that activates the Android native DND mode, responds with an auto-reply message with the receiver's activity information, and lets the sender choose when to alert the receiver by typing the command. The Slack and Messages DND mode and this system design are similar with MyButler in terms of its concept of status sharing and sender-side notification control. However, they provide the functionality only for unavailable situations when the user does not want to be 'disturbed'. The DND mode also by default mutes all notifications from everyone. On the other hand, the functionality in MyButler is open for situations other than unavailable ones and takes no effect on notifications from people other than MyButler partners. MyButler therefore allows more freedom for the types of status to share with senders other than 'Do Not Disturb'. The various types of status might also contribute to making the sender's choice to send a notification more flexible, not constrained to urgent matters. We explore with MyButler what types of status people share privately with their close friends or partners and how these affect the sender's choice of notification.

2.2 Increasing Social Awareness among Remote Users

Previous research investigated techniques to sense and provide a user's contextual information (e.g., location [3, 11, 31, 46, 49, 59, 64], motion [4, 22, 31, 49, 64], mood [22, 49], and social setting [46, 49]). These contextual cues were designed to assist users in determining their friend's presence and availability in telecommunications by contextualizing the phone contact list [3, 18, 31, 46, 53, 56] and the friend list of instant messengers [4, 11, 22, 49, 59, 64]. De Guzman et al. [10] studied among the readily sensed contextual cues that are *actually needed* for callers and callees. Oulasvirta [44] explored design issues from individual awareness cues to product concepts.

One line of research focuses on intimate relationships and examines their practices of sharing contextual information. Researchers have investigated sharing "single, ephemeral streams" [16] of contextual information such as location [2, 3, 37, 52], ambient sound [36], motion status ("moving" or "not moving") [5], calendar schedules [60, 61], and heart rate [20]; and multiple streams of contextual information (e.g., steps, location, battery level) [16]. The impact of increased contextual awareness on the relationship are many-folds. Sharing contextual information changes the communication dynamics by increasing or decreasing direct communication [2, 16, 20, 52]; facilitates coordination [2, 5, 16, 52, 60]; provides a better understanding of each other [5, 16]; and evokes feelings of connectedness [2, 5, 16], empathy [20], and peace of mind for knowing the partner's safety [2, 5].

We explore context information sharing with close ties to the MIM experience by studying the impact of sharing custom statuses with close relationships in response to a message.

2.3 Mitigating Interruption

As a way to reduce smartphone interruptions at inopportune moments, researchers have studied context-aware notification systems that analyze the message receiver's situation and the value of notifications to regulate the notification delivery. Based on the contextual data, previous work estimates interruptibility of a user and defers notifications until opportune moments [12, 21, 28, 41–43, 45, 47] or filters irrelevant notifications [13, 38]. Mobile sensing is often used for understanding the receiver's situation through the information such as location [47], sound intensity [45], physical activity [21, 45, 47], and mobile phone activity [12, 41–43, 45]. However, mobile sensing techniques are not sufficient to achieve a comprehensive level of context sensing; for example, if a user is in a dorm room and the phone is placed still, it is hard to distinguish whether the user is studying, relaxing, or getting ready to go out.

The content of a notification also plays an important role toward a receiver's receptivity [13] and acceptance to disruptive notifications [40]. Nevertheless, due to privacy concerns, the value of a message notification has to be analyzed only using the type and title of notification instead of the message content [39]. Leveraging the social characteristic of mobile messaging notifications, MyButler compensates the deficiency of mobile context sensing by involving the message sender to actively participate in the notification control. It could be beneficial to include the message sender in the loop for notification control, who knows the content of the message and understands the urgency of the message. In this way, the system does not need to see the private content of the message notification.

3 DESIGN CONCEPTS OF PRIVATE STATUS SHARING AND SENDER-CONTROLLED NOTIFICATIONS

We present our two design concepts, private status sharing and sender-controlled notifications, with explanations on our design rationale.

3.1 Private Status Sharing

The three characteristics of our private status sharing are personal, custom, and reactive status setting and sharing.

3.1.1 Personal Sharing. Private status sharing allows users to select their own personal partner(s) to share the status with. This aims to overcome the concerns of privacy invasion in existing systems where anybody on the user's messenger friend list can see the user's status. Being connected to any 'friends' on the messenger has been pointed out as a source of stress [54]. We expect that our personal and targeted sharing design brings status sharing behaviors that are different from the status setting features in commercial MIM apps such as KakaoTalk's and Slack's *Status*, and WhatsApp's *About*. Although not deployed in our field study as our participants are in pairs, it is possible to set different status for different partners (e.g., "busy working" to the manager and "available" to the significant other).

3.1.2 Custom Status Setting. The status for sharing is fully customizable and manually selected. Custom status setting supports high degrees of expressiveness of one's status and also allows room for *plausible deniability* [1] and *Butler Lies* [51], which are crucial in alleviating social pressure [48]. The manual selection targets to mitigate the concerns related with automated availability indicators, e.g., feeling of being observed and patronized and false expectations due to wrong prediction [48]. Since manual selection has a drawback of having to frequently update the status [4], our design incorporates two components to ease the burden of manual updates; *Shortcut* and *Favorite Status List*. The *Shortcut* is designed to shorten the sequence of status updates in existing MIM apps by adding an external shortcut that directs the user to the status update interface. *Favorite Status List* is a list of customized statuses managed by the user that eliminates the burden of typing frequently used statuses every time for an update.

3.1.3 Reactive Status Sharing. The configured status is made available for the partner only when the partner sends a message to the user, as shown in Figure 2 (c). This has two expected benefits over existing mechanisms that always keep the status in the user profile or on the contact/friend list. First, it is intended to reduce the feeling of being observed. The status is disclosed only when the friend actually has the intention of talking to the person instead of spying on someone's status. Reactive disclosure is discrete while existing always-on disclosure is continuous. Even when someone tries to send a message only to find out the user's status, the user becomes aware of who has seen their status at what time. Another benefit is that the status is noticeable. Because the pop-up notification appears when one sends a message, it attracts the sender's attention.

3.2 Sender-Controlled Notification

Sender-controlled notification is the concept of involving the message sender to decide how to process the notification for their own mobile instant messages. Two basic options are whether to generate the notification (*Alert Now*) or not (*Don't Alert*). *Alert Now* is the same as what an unmodified mobile instant message would generate. *Don't Alert* is sending the message without leaving any pop-up, sound, or vibration notification to the receiver. More advanced options could be added such as alerting when the message receiver becomes available or after a defined time period. For the scope of this work, we analyze the concept of sender-controlled notifications only with two basic options.

For the design of sender-controlled notifications, we had two candidates: Messages-like instant auto-reply message (on the right of Figure 1) and notification-based auto-reply pop-up (Figure 2 (c)). We chose the latter based on the following two criteria.

3.2.1 Notification method selection must be unobtrusive. In many occasions, MIM users send a group of consecutive messages rather than a single message. The best timing to trigger the notification selection panel is ideally at the end of the group of consecutive messages. However, with the current technology, it is impossible to predict correctly when the sender is finished with their messages in mind. Therefore, we display the notification selection panel along with the receiver's status after the first message has been sent and wait for the sender's selection as shown in Figure 2 (c). Since our system works similar to an auto-reply, one possible design choice was showing the notification selection panel in the form of a usual KakaoTalk message. However, our pilot study showed that such message-format quick auto-reply surprises the auto-reply receiver (i.e., the message sender) as it looks like the person has immediately replied.

3.2.2 Notification method selection must be easy to operate. Considering the amount of instant messages people send every day, the selection must be simple, intuitive, and easy to operate as it would otherwise impose significant burden on user input. Messages-like auto-reply has the benefit of greater deployability as its operation requires installation only on one side. However, the user has the burden of typing the command and pressing the send button. On the other hand, the user input required in our notification-based auto-reply pop-up is pressing only one button.

4 MYBUTLER: AN ANDROID APP PROTOTYPE OF PRIVATE STATUS SHARING AND SENDER-CONTROLLED NOTIFICATIONS

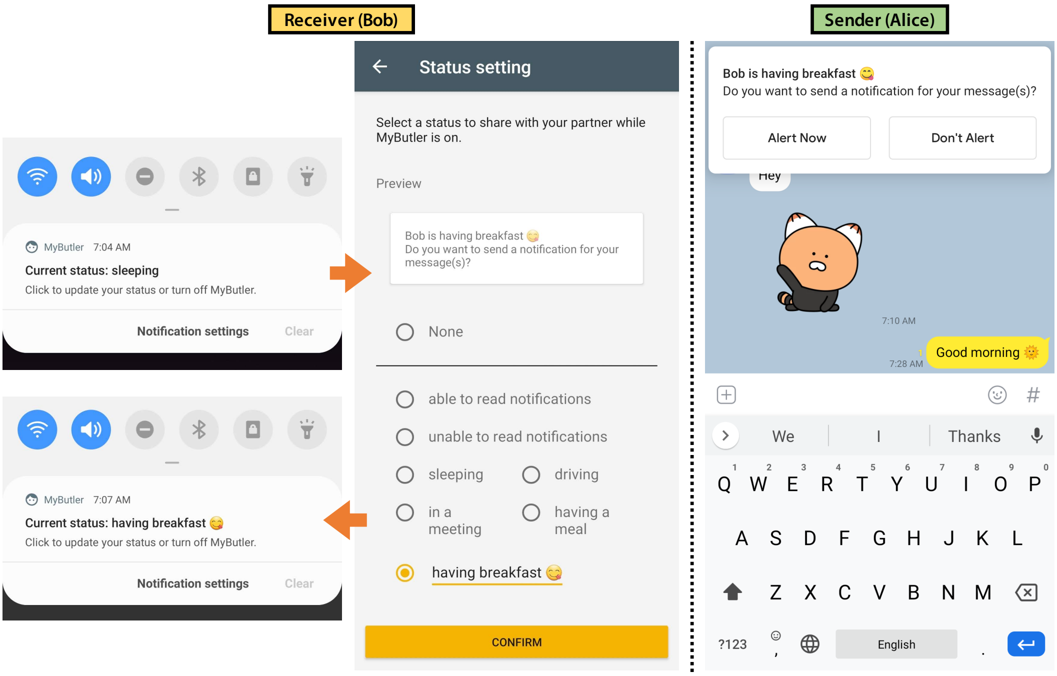
We apply the design concepts of private status sharing and sender-controlled notifications to building an Android app prototype *MyButler* that works with KakaoTalk, a popular commercial MIM app in Korea. We chose KakaoTalk as the target MIM app because it is a dominant messaging app accounting for 94.4% of the total time spent on mobile messengers in Korea where the user study was conducted [29]. Our primary purpose of building the prototype is to observe the impact of our two design concepts on users' natural, personal messaging behavior. KakaoTalk-friendly implementation is therefore more advantageous over building a standalone MIM app or utilizing other more programming-friendly MIM apps such as Slack because switching the messaging environment would inevitably bring a new factor to the user's messaging experience. MyButler could be easily modified to work with other MIM apps as well.

We demonstrate how MyButler works with a walkthrough of an example scenario and explain the implementation details.

4.1 MyButler Walkthrough

MIM communication is a bidirectional progress in which the two parties of a conversation take turns to send and receive messages from the other. For the convenience of explanation, we take a look at one direction of the communication in this walkthrough: Alice is sending a message to Bob. MyButler works in the following steps:

(1) Alice and Bob both install the MyButler app and add each other as the MyButler partner using their login email address.
(2) Bob opens the MyButler app or uses the Android notification bar *Shortcut* (Figure 2 (a)). *Shortcut* resides in the notification drawer and thus acts as a reminder for status update as well.
(3) Bob selects a status from his Favorite Status List or enters a new custom status in the status setting interface (Figure 2 (b)). In the figure, Bob updates his status from ‘sleeping’ to a new status ‘having breakfast’. Selecting a new custom status and pressing the Confirm button triggers a dialog that asks for adding the new status to the Favorite Status List. If Bob chooses



Current status: sleeping
Click to update your status or turn off MyButler.

Current status: having breakfast 😋
Click to update your status or turn off MyButler.

Select a status to share with your partner while MyButler is on.

Bob is having breakfast 😋
Do you want to send a notification for your message(s)?

○ None
○ able to read notifications
○ unable to read notifications
○ sleeping ○ driving
○ in a meeting ○ having a meal
● having breakfast 😋

Bob is having breakfast 😋
Do you want to send a notification for your message(s)?

(a) Shortcut to status setting

(b) Custom status setting

(c) Private status sharing & sender-controlled notification options

Fig. 2. The usage flow of MyButler, an Android app prototype that instantiates our two design concepts of private status sharing and sender-side notification control. Bob the message receiver can use (a) the shortcut in the notification drawer, which leads to (b) the status setting interface that has a selectable Favorite Status List and a custom status option at the bottom. Tapping the Confirm button updates the status in the shortcut accordingly. When Alice sends a KakaoTalk message to Bob, she will see (c) a pop-up notification saying 'Bob is having breakfast' with two notification options for her messages, *Alert Now* and *Don't Alert*.

to add, the next time Bob updates the status, he will see 'having breakfast' in the selectable list without having to type it again.

(4) Alice sends a KakaoTalk message to Bob and receives a pop-up notification about Bob's status along with two sender-controlled notification options, *Alert Now* and *Don't Alert* (Figure 2 (c)).
(5) Before selecting an option, Alice sends all KakaoTalk messages that she intends. After sending the last message, Alice selects a notification option.
(6) If Alice chooses *Alert Now*, Bob gets a push notification for a group of messages that Alice sent in Steps (4) and (5). If Alice chooses *Don't Alert*, Bob gets no notification and can read the messages when he opens the KakaoTalk app later at his convenience.

Note that there is no default alert option for Alice, and thus Alice must select either *Alert Now* or *Don't Alert*. This was a methodological decision to avoid potential bias in probing user behaviors when using sender-controlled notifications.

4.2 MyButler Implementation

MyButler is implemented as an Android application and is built on top of the open-source Notification Log framework [62]. It utilizes Android `NotificationListenerService` to extract metadata and logs from Android notifications; it requires the permission for Notification Access. Based on the notification metadata (package name and title), the receiver-side app detects the arrival of a KakaoTalk message from a MyButler partner and triggers the pop-up notification on the sender side. The receiver-side and sender-side MyButler apps communicate through Firebase Realtime Database [15] and Firebase Cloud Messaging [14].

To prevent the generation of KakaoTalk's own push notifications and implement sender-controlled notifications, MyButler requires the notifications for the specific 1:1 chatroom to be muted in KakaoTalk beforehand (KakaoTalk supports chatroom-specific notifications settings). When muted, the notifications will appear in the notification drawer but without sound, vibration, or visual push. On message arrival, MyButler saves a copy of the notifications for later use and cancels these notifications. If the sender chooses the *Alert Now* option, the receiver-side app will fire the copied push notification, containing the message(s) from the sender. For the *Don't Alert* option, as the original message notifications were already canceled, MyButler does not take any further action.

5 FIELD STUDY

We conducted a field study with 11 pairs (five romantic couples and six friend pairs) to investigate in-situ user behaviors of using MyButler. Our field study aims to understand the following:

- How MIM users express their status through private status sharing,
- What roles the privately shared statuses play in MIM communications, and
- Why MIM users choose to alert or not alert for their messages.

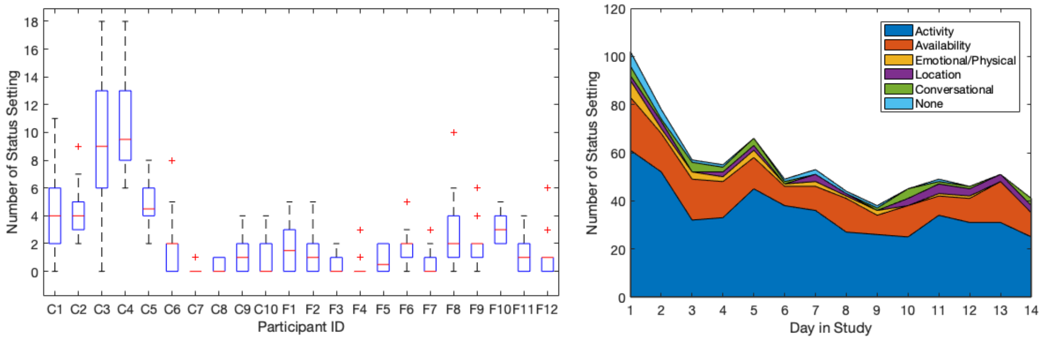
5.1 Participants

We recruited 11 pairs of friends and couples (female=8, male=14; mean age=22.9, min age=18, max age=29) who use Android phones and chat with each other on KakaoTalk more than three days in a week. We targeted friends and couples as the focus group for our study because they are the most active user groups of WhatsApp [8] and have relational needs for mediating awareness (e.g., [16, 19]). Five couples (all female-male couples) and six friends pairs (four male-male pairs, one female-male pair, and one female-female pair) voluntarily signed up as a pair for the study participation. Participants include 12 undergraduate students, six graduate students, three office workers, and a middle school teacher. The relationship length varied from three months to eight years. Among couples, one was married. There was no cohabiting pair; the married couple lived apart during the weekdays for work and together only in the weekends. In the rest of the paper, we refer to our couple participants by C1-C10 and friends participants by F1-F12. Participants were recruited through a university online forum and social networking services. We paid 50,000 KRW (approximately 41 USD) to each participant for participating in the study. This study was approved by the Institutional Review Board of the authors' institution.

5.2 Procedure and Data Analysis

Participants were first invited to an introductory session for MyButler app installation and tutorial on how to use the app. They were asked to answer questions related to their ordinary smartphone ringer mode and MIM app notification settings. They downloaded and installed our Android app on their own smartphones. The tutorial session involved a walkthrough of our app: turning on/off MyButler, changing the status, selecting the notification choice after sending a KakaoTalk message, and a demonstration of how it looks on the other side when each notification choice is made. Any usage logs collected during this tutorial session were excluded from data analysis.

We instructed participants to freely use MyButler for 14 days with their partner. They were told to turn on/off MyButler and update their status as they wish. We also explained that if there is



(a) Daily Number of Status Setting Events Per Participant

(b) Number of Status Setting Events Over Time

Fig. 3. (a) A boxplot of daily status setting occurrences per user; (b) A stacked area plot of the number of status setting events of all users for each category over the course of the study.

no change in status for longer than a day, the app will assume the user has forgotten to update and send a reminder notification for status update. We discuss its possible side effects on the study results in Section 8.5. Note that MyButler's functionality is only applied to the study partner, and messages and notifications remain unaffected for communication with others. The usage metadata of status update logs and sender's notification choice logs were stored in our cloud storage through our app with participants' permission. We also discuss its possible side effects in Section 8.5.

After two weeks of usage, participants joined an interview session. The interview session took place within 1 to 12 days after the usage period. The session involved a 50-minute pair interview and 5-minute individual interviews. A timeline visualization of the pair's status updates and notification choices was used as an interview reference to revisit their usage history. Participants also looked back on their KakaoTalk chat on their own smartphone for a better recall of a situation during the interview. We first interviewed both participants together to help them better recollect the context of conversation and availability at the time. We observed that participants could crystallize the context by combining their parts of memories. During the 5-minute individual interviews, we asked questions that could hurt the partner's feelings such as cases of butler lies or deceptive behaviors. We also asked if they wanted to talk about other things that they could not disclose in the pair session.

The collected metadata and interview data were analyzed through the procedure of thematic analysis [6]. The interviews were conducted and transcribed in Korean. The lead author extracted meaningful mentions from the transcripts and coded them. The other authors verified the coded labels and sorted quotes by themes. The quotes used in the paper were translated into English by bilingual authors. We present the field study results in Section 6 and Section 7.

6 PRIVATE STATUS SHARING PRACTICE

We present field study results of the private status sharing usage and findings from the participants' perceptions of private status sharing.

Over the two weeks study, 22 participants updated the status with the average of 35.18 times per user (SD=37.77, min=3, max=148). Figure 3 (a) presents a boxplot of daily status setting occurrences per user. The average of daily status setting occurrences is 2.51 times per day per participant (SD=2.70, min=0.21, max=10.57); 3.77 times for couples (SD=3.57, min=0.21, max=10.57); and 1.46 times for friend pairs (SD=0.92, min=0.36, max=3.29). Figure 3 (b) illustrates the number of

Cluster	Occurence	Cluster	Occurrence
Available	164	Washing	24
Sleeping	98	Exercising	23
Having a meal	60	On the move	23
Driving	33	* Doing an experiment	19
Resting	28	* In bed	19
Getting ready	26		

Table 1. The top 10 most updated status clusters. Statuses with the same semantic meaning were grouped into status clusters through open coding. Note that there are 11 clusters because the statuses with an asterisk (*) were a tie.

status setting events for each day. The status setting practice was the most active on the first day of the study (102 times), decreased after the first two days, and stabilized until the end with the least active day being Day 9 (38 times). The average number since the third day (49.16 times) is about half of the first day. The participants agreed that they used more frequently at first, indicating signals of a novelty effect. After the second day, there were ups and downs in the number of daily status updates, but without drastic fluctuations (SD=7.76).

6.1 Six Status Categories

Through MyButler's custom status setting, participants expressed themselves using 12.59 different statuses per user on average (SD=10.05, min=4, max=42) and a total of 210 different statuses as a whole. Two authors independently coded every status and grouped the 210 statuses into clusters of the same semantics. The two authors reviewed the codes and clusters generated by each other and discussed the results until reaching a consensus. The bilingual authors then gave an English name for each cluster that represents the given codes. As a result, for example, the statuses 'sleeping', 'going to sleep', and 'zzz' are clustered to Sleeping; and 'able to receive notifications', 'able to check my KakaoTalk often', and 'able to get in touch' are clustered into Available. This semantic clustering resulted in 69 semantic status clusters and Table 1 presents the top 10 most occurring status clusters.

We further categorized the 69 clusters into six categories: Activity, Availability, Emotional/Physical, Location, Conversational, and None. The first author clustered statuses by their codes and extracted the common features to develop these categories. The other authors reviewed them and resolved discrepancies by discussion. Figure 3 (b) shows the number of status setting events for each category over the course of the study.

- **Activity** statuses indicate the current activity or task of the user, e.g., Sleeping, Driving, Having a Meal, and Getting Ready. Activity statuses are the most frequent category accounting for 64.3% of all status updates. Among the top 10 most occurring status clusters (Table 1), all except for the Available cluster belong to this category, and Sleeping is the most popular from the Activity category. One participant said: *"I used the sleeping status often because sleeping particularly has a definite beginning and end. For the others, I updated whenever it came across my mind"* (F1).
- **Availability** statuses directly communicate one's (un)availability and (un)willingness to receive notifications. There were some uses of a combination of Activity and Availability such as 'don't wanna look at the phone and lying in bed [heart emoji]' and 'Hearthstone[1] (don't

[1] It is an online card game.

interrupt)'. Availability statuses are the second most frequently set category accounting for 23.3% of all.

- **Emotional/Physical** statuses express emotions or physical conditions of the user, e.g., 'happy', 'angry', 'hungry', and 'hungover'. This category contributed 3.2% of total statuses updates.
- **Location** statuses are the name of the location such as 'in the restroom', 'in the lab', and 'at the airport', accounting for 3.7% of status updates.
- **Conversational** statuses resemble casual, instant messages in a conversational tone. Participants leveraged the status prefix '*User* is' and *"tried to make the status a complete sentence"* (F9) to sound more conversational. A pair of friends threw jokes at each other, e.g., '[*User* is] cooler than *the partner*'. Couples used affectionate expressions such as '[*User* is] missing *the partner*'. When one user changed the status to a Conversational status, the partner tended to reply by setting their own status to a reciprocal Conversational status as well.
- **None** status is the option for not showing any status but still activating the sender-side notification control feature. Participants set the None status when they were not doing anything significant and did not want to prevent the other person from contacting them.

6.2 Roles of Statuses

Through the interview, we found that statuses play various roles in MIM communications.

6.2.1 Relieving the burden and expectation of immediate response. Participants changed their status to notify the sender of the reason for unavailability to attend to notifications or reply to messages at the moment: *"I updated the status to let the partner know that I was exercising so won't reply to him"* (F4). 'Sleeping', 'having a meal', 'driving', 'playing games', and 'in a meeting' are other examples of statuses set with the intention of explaining their unavailability. Participants preferred to provide a specific Activity context to be more explicit than setting 'unable to read notifications' of the Availability category because *"It looks too unkind"* (F2) and *"The partner will be curious why I can't read notifications"* (C3). Accordingly, 'unable to read notifications' was only selected a total of ten times, combining all participants' data whereas 'sleeping', 'having a meal', 'driving', 'playing games', and 'in a meeting' were selected 98, 60, 33, 23, and 10 times, respectively.

By sharing their unavailable status, participants felt less obliged to respond immediately to messages. F3 said,

> *"I'm one of the people who reply to even the smallest things, but when I updated my status, I was less concerned about not answering. When I was doing an experiment, I could only use one hand, so I checked the message but couldn't be bothered to type. I read it and felt that I don't have to write back because I already informed him of my status. It was more convenient, and it made my life easier. I could just end it after reading the message. I felt less obliged to answer."*

Given the receiver's unavailable status, the sender did not wait for an immediate reply. F1 said,

> *"For example, when I send 'What are you doing?', if I didn't have this app, I wouldn't know when he'd reply. It's an infinite probability. When I have to work on something else and I'm on my phone, I'll stay on the phone if I know he'll reply soon, and otherwise put it away. When I have to make that kind of decision, if his status says 'having a meal', I can easily decide to put away the phone and not wait."*

The shared understanding of unavailability between the pair eased the burden of immediate response on the receiver's side and reduced the effort of waiting for an unpredictable reply on the sender's side.

6.2.2 Advertising Availability. Participants expressed their availability by setting the status to 'able to read notifications' instead of explaining why they are available in contrast to unavailability cases. This status was selected 164 times in total, which is the most frequently selected status of all. The high frequency is related to participants' popular status setting strategy of using 'able to read notifications' as a default status option for any trivial activities: *"I updated the status for activities that are not too trivial to share, such as 'going home' and 'working part-time'. For any other cases I set 'able to read notifications'"* (F1). This can be interpreted as a result of not including a default option for the explorative purpose of MyButler.

6.2.3 Real-Time Status Sharing. Participants, especially couples, used the status to reflect their real-time status that the partner might be interested in. Participants who used MyButler for this purpose liked the aspect that it simplifies the process of real-time status sharing, which is a common communication practice by couples. Couples tend to talk about what they are currently doing through instant messages. In comparison with other communication channels, MyButler provides a quicker and easier way to keep up with the daily routine of each other through a few taps while other methods require the effort of typing a well-formed message. Participants found it helpful especially when they were in a rush (e.g., 'getting ready'):

> *"I often fall asleep or fail to wake up when I have to get ready to meet my boyfriend, so he often gets nervous and calls me a lot to ensure that I'm up. With MyButler, he could stay more relaxed in these situations when my status says 'washing' or 'getting ready'. There are these cases where you have to get ready in such a rush that you don't even have the time to open KakaoTalk and type the message. With MyButler I just have to tap the [status setting] button and take a shower"* (C2).

Another example was when the participant was in inappropriate moments for texting:

> *"I tell my boyfriend to not KakaoTalk when he walks to the dorm because it's dangerous to look at the phone while walking at night. Sometimes I'm worried if he's alright but I can't send a KakaoTalk message because it'd be dangerous. In these cases, I was relieved when I saw his status updated [from 'walking back to the dorm' to 'lying in bed']"* (C2).

This echoes the literature that couples use the shared contextual information such as motion [5] or location [2] to check on one's safety and get peace of mind. Although MyButler employs manual status sharing, for participants who have built up the behavior of updating the status to the level of real-time, it induced similar feelings of peacefulness as the auto-updated motion or location information.

6.3 Personal Preferences on Reactive Status Sharing

The biggest difference between MyButler and existing status sharing systems is that MyButler offers reactive status sharing, only in response to a message. Existing systems, on the other hand, always display the status on the user's profile or the friend list. Our participants had varying preferences over reactive status sharing for the following reasons.

6.3.1 Seeing the status requires sending a message. The reactive status sharing takes effect only when the sender has sent a message. Some participants who disliked reactive sharing wanted to be aware of the partner's status without sending a message: *"I set my phone to silence most of the time, so getting alerts didn't mean a lot to me. I liked that I don't have to reply to the message to inform him of what I'm doing. It'd be better if he can know this even without sending a message"* (F8). On the other hand, participants who preferred MyButler over existing systems valued MyButler for allowing senders to not have to defer sending messages. F7 said, *"If my status is set to 'in a meeting', [in existing systems] the other person would see my status and decide not to send a message. He could* *check my status again later and send that message, but he could also forget. It's better for me if he sends the message anyways and then learns that I might reply later because I'm in a meeting. This way, I don't miss any messages."*

6.3.2 Sharing with people who are actually close. Manually tracking the list of people to share the status with is the most secure practice of status sharing, as adopted by Instagram's Close Friends. But it can be bothersome when the list gets longer. Reactive status sharing could be an alternative in a real-life scenario with a broad range of online friends. MyButler can support private sharing with people who are *actually close*, or people who they actually talk to. C2 said, *"People who are 'actually close' are those who keep in touch. Sharing my status doesn't become privacy invasion when only these 'actually close' people can see my status. If the range of sharing gets broader, it gets more uncomfortable to share my status."* Participants also found reactive sharing to be *"more aligned with today's trend to share my status only with people who want to talk to me rather than anybody"* (F3). When used in a broad range of group, there could be a malicious user who abuses the reactive sharing by sending a pointless message to probe the status information. However, it is highly suspicious if a distant user consistently sends a message without a specific purpose, and thus the receiver can easily detect and block the suspicious user.

6.3.3 Pop-up appears frequently. Some participants disliked the pop-up notifications with statuses and sender options that frequently appear in response to their messages: *"It's a little annoying. I think it's a little uncomfortable for people who use KakaoTalk a lot. We literally 'KakaoTalk' all the time. It feels like making a double effort because it keeps popping up, and I have to keep selecting"* (C6). The current version of MyButler generates a pop-up notification with the status information and alert choices every time the sender sends the first message after the last alert choice. As mentioned in Section 4.1, the current version does not provide a default option for alert choices to avoid potential bias in probing user behaviors. Instead, generating the pop-up only when the receiver's status has changed could be one approach to reduce the pop-ups.

6.4 Perceived Burdens of Status Setting and Suggestions for Partial Automation

Some participants felt that the process of status setting in MyButler is simple to use for real-time status sharing. However, other participants, especially friends, felt it was rather cumbersome and often forgot to update the status. One of the friends participants said, *"I don't think I'll be bothered if I can't get in touch with him. There's nothing to worry about. If it were my girlfriend, I would be. But for him, even if I don't answer for hours, he'll just think I might be busy, and that's it. I don't think I have to put up with the effort [of status setting] for a friend"* (F8).

Participants suggested several ways to ease the burden of manual status setting such as one-touch buttons in the home screen or adding automation rules. C6 said, *"I'd like to have image icons in the home screen like a dog icon [to set the status 'playing with Nana (her dog)'] and a sleeping icon [to set the status 'sleeping']"*. Smartwatch support for such one-touch buttons was also suggested by F8. Participants also expressed a desire to customize automation rules that utilize users' habitual smartphone actions as triggers. For example, F8 suggested using a particular Wi-Fi or Bluetooth connection as the trigger in a similar way with the Android app Automate [35]. F3 also said, *"I always set the alarm before going to bed. It'd be great if this can be automatically mapped to setting the 'sleeping' status. I never forget to use a service like the alarm clock because it's necessary for me. It would be useful if the status setting is linked to such services I use out of necessity."*

Another approach is leveraging the calendar or schedule items that are carefully managed by many users, especially professionals. Participants articulated concerns for unwanted exposure of their schedules, so they wanted the ability to cherry-pick which calendar items are shared in which form. C1 raised another concern that simply copying-and-pasting the calendar item on the status would lose the emotional aspect of MyButler: *"In our case, we maintained our romantic tone in the statuses as well, but calendar items are usually written in official terms like 'lab meeting'. If it's possible to translate from these official terms into a more friendly tone, it would preserve its 'social' aspect."* This idea of the personalized tone adjustment can be applied to not only calendar items but also other automated status settings.

Participants emphasized the need for complete control due to the concerns of privacy and possible social conflicts raised by the status. One of the couples participants said, *"Automation is okay for only the statuses that I'm not ashamed of. For example, 'doing homework' or 'in a meeting' are okay to be revealed, but something like 'going out for a drink' would be dangerous"* (C10). A participant also preferred a confirmation-based method that receives the user input for confirming a status update, for example, by asking "Do you want to change your status to 'watching YouTube'?".

6.5 Butler Lies and Plausible Deniability

Previous research implicates that instant messaging system designs should allow plausible deniability [1] and butler lies [17, 51]. Such system would leave room for which the user can blame the system for their behaviors against social expectations in messaging so that the user can maintain a positive image of themselves and sense of their own autonomy [17] in their interpersonal communication.

Through our user study, we discovered three types of butler lies, or intentional deceptive behaviors, in status setting. The first type is leaving an outdated status unchanged: *"I intentionally kept my status 'watching movie' unchanged even after I was done watching the movie. I did so because I didn't want to be bothered. On one side I couldn't be literally bothered to manipulate the app, but on the other side, I just didn't want to let him know that I could get in touch"* (F2).

The second type is changing the status prior to the upcoming event. For instance, participants reported of setting the status to 'sleeping' when they were not going to sleep right away but wanted to be free of messages at night. F10 said, *"If I don't change the status [to 'sleeping'], the conversation will continue. I usually sleep at 3 or 4am, but I change the status [to 'sleeping'] around midnight and play with the phone, and watch YouTube. KakaoTalk messages really bother me when I'm doing that."*

The third type is lying about the current status: *"When I wanted to have free time for myself rather than chatting with my girlfriend, I set my status to 'watching TV' and did whatever I wanted to do. [...] I thought she would understand even if I reply later because my status says I'm 'watching TV'"* (C10).

7 TO ALERT OR NOT TO ALERT

We now analyze the in-situ usage of sender-side notification controls (*Alert Now* and *Don't Alert*) and participants' perceptions on this feature.

Participants made a total of 2,922 *Alert Now* (74.6%) choices and 994 *Don't Alert* (25.4%) choices over the two weeks of study. Figure 4 shows the number of *Alert Now* and *Don't Alert* choices that each participant made. For most participants, *Alert Now* was the dominant option, and participants explained they regarded this as the default option unless they had other reasons to choose *Don't Alert*. However, the F11-F12 pair used *Don't Alert* as their default unless they had special reasons for *Alert Now*. We first discuss the first type of behavior in Sections 7.1 and 7.2 and the second type in Section 7.3. Figure 5 shows the percentage and number of the two notification choices for the top 10 *Don't Alert*-ed (Figure 5 (a)) and top 10 *Alert Now*-ed (Figure 5 (b)) status clusters.

7.1 Why "Not Alert"?

Don't Alert accounts for 25.4% of all sender-side notification choices. We present the key reasons for choosing to mute notifications for their messages.

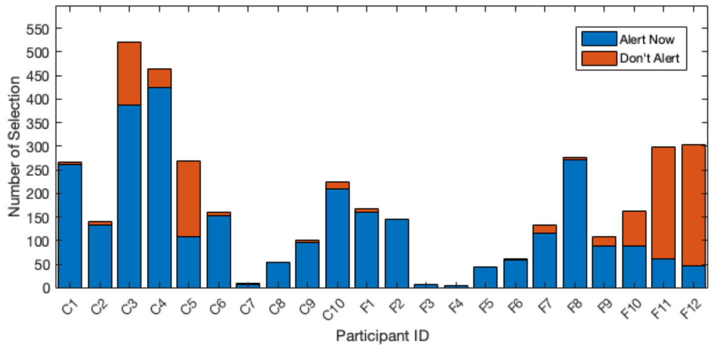
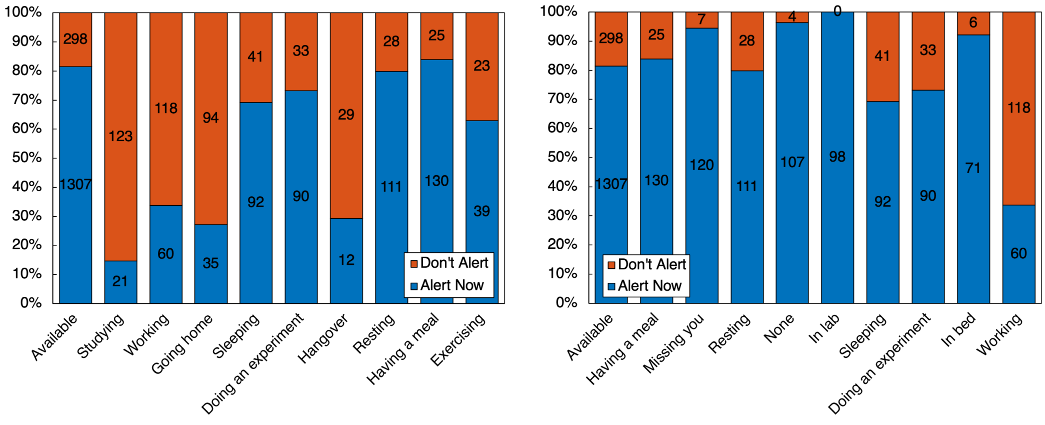


Fig. 4. Alert Now and Don't Alert statistics for each participant.



(a) Top 10 *Don't Alert*-ed Status Clusters

(b) Top 10 *Alert Now*-ed Status Clusters

Fig. 5. Percentage and number of sender-side notification control selection for (a) top 10 most *Don't Alert*-ed status clusters and (b) top 10 most *Alert Now*-ed status clusters. The area of the bar represents the percentage of each choice for the status cluster, and the number inside the bar represents the number of selection.

7.1.1 *Prevent Interruptions by Sound/Vibration at Inopportune Moments.*

Some participants tried to avoid interruptions by the sound or vibration of a message notification when the receiver's activity was set to 'sleeping', 'driving', 'on the move', 'in a meeting', etc.: *"I was afraid the notification would wake him up, so I left the message and chose Don't Alert"* (C9). While participants did not send notifications to avoid interruptions, they used MyButler to still leave the message as a sent receipt: *"I liked it that when the partner was in a meeting and I didn't want to interrupt, but I wanted to let him know that I replied with sincerity."* (C2). When participants recognized that their partner chose *Don't Alert* in the inopportune moments, they thought of the partner as being considerate: "I didn't know he was this considerate" (F10).

Note that the perceived level of distraction by MIM notifications and expectation of the receiver's ringer mode vary among participants. The differences in perception contribute to diverging notification choices even for the same receiver status. For example, when the partner was asleep, F10 opted for *Alert Now* because she thought it would not "force making a sound or vibration" against her partner's ringer mode settings (Section 7.2.1). However, C9 chose *Don't Alert* as she "was afraid the notification would wake him up".

7.1.2 Hide Messages from Other People Around. Participants also utilized *Don't Alert* when they did not want their messages to be seen by other people around the partner: *"When she was with family or friends, they might see the content of my messages on her phone. I didn't alert in that case because I didn't want other people to see my messages."* (C10). This reason also explains why C5 has an exceptionally large number of *Don't Alert* compared to other participants as shown in Figure 4. His wife (C6) is a secondary school teacher, and *"her mischievous students sometimes just pick up her phone and read the KakaoTalk notifications"* (C5). C5 did not alert any messages when C6's status was set to 'working'.

7.1.3 Sender is Unavailable or Unwilling to Talk. Even if the receiver's status indicated availability e.g., 'able to read notifications' and 'bored', participants did not send notifications when the senders were unavailable at the moment. F7 said, *"I was in the middle of organizing my office. I sent this message at 1:02 PM. I thought I'd be physically too busy to reply, so I chose to not alert. After I was done organizing, I sent another message at 4 PM."* Participants also skipped sending notifications when they were unwilling to get an immediate reply: *"It was just because I didn't want to get the message. I didn't want to be bothered getting the reply. [...] That's why I didn't read it for almost an hour"* (F8).

7.2 Why Alert?

Alert Now was the dominant notification choice by senders, accounting for 74.6% of all choices. We summarize the key reasons for choosing to send messages *with* notifications.

7.2.1 MIM Notifications Do Not Cause Huge Distraction or Pressure. Some participants reported that they did not perceive MIM notifications would be a major source of distraction to their partner in a close relationship: *"I thought KakaoTalk was fine. I would hold myself from calling, but he'll see KakaoTalk messages whenever he is able to. I alerted the message so he can check the notification later, not necessarily now"* (F5). They also trusted that the receiver would have already configured their ringer mode settings appropriately in necessary conditions: *"When he was watching a movie at the cinema, he would have certainly put off any source of smartphone distractions."* (F1). F10 also said, *"Alert Now doesn't force making a sound or vibration [against his ringer mode settings], so I thought it wouldn't matter even if he's asleep."*

Accordingly, when receiving *Alert Now*-ed messages, participants did not feel distracted because they were *"already familiar with receiving notifications in whatever situations"* (C10) from their intimate partner. Participants also did not perceive *Alert Now* as a source of pressure for a quick reply even though they were cognitively aware of the fact that they received the notification because the sender explicitly selected *Alert Now*. C2 said, *"We both selected Alert Now when we were playing League of Legends. I was implying that 'I'm sending a reply now, but I do understand if you don't reply back immediately and take your time for the game'. We implicitly had a mutual agreement on this."*

7.2.2 "I Want My Messages to Be Quickly Noticed". Participants who never or rarely chose *Don't Alert* wanted their messages to be seen as quickly as possible. F2 who never selected *Don't Alert* said, *"I thought it'd be great if he can see my messages quickly and I didn't really understand why we have the Don't Alert option anyways."* Some participants used *Alert Now* when they wanted a quick response from the partner. For example, F7 mostly chose *Don't Alert* when the partner was 'in a meeting'. However, he chose *Alert Now* for important or urgent messages that require an immediate response even in the same situation: *"This was an important one. This was for scheduling today's interview. I knew he was 'in a meeting', but I needed to get back immediately with the time"*. Another case for wanting a quick response was when participants were bored and wanted to chat. F9 said, *"I sent notifications when I was curious about what she was doing. I wanted to receive responses right away when I was bored."*

7.3 Using Don't Alert as the Default

F11 and F12 showed a different pattern from other participants as they used *Don't Alert* as their default notification choice if not for special cases. F12 expressed the lack of necessity to notify non-urgent messages: *"We do a lot of pointless chitchats on KakaoTalk, so we rarely chose Alert Now and use it only when we needed an immediate response"* (F12). Some examples of the alerted messages were decision-making for the dinner menu; sharing information about a tutoring job recruitment that usually gets taken quickly; and sharing a job offer result. The pair reported that the notification choice made by each other reciprocally affected the decision and gradually built a consensus: *"I think I've got a sense of what she thinks is important and what she feels is not important by looking at which messages come with notifications and which ones don't"* (F11).

The pair noted that their usage pattern helped them check important messages in a more timely manner. F12 said, *"I usually have a huge stack of notifications, so I don't care if I get notifications or not. With this app, it felt like I get the notification only when it's important."*. F11 added, *"I also liked that. There's no alert when it's not important. I get alerts only when it's important, so I can check the message more quickly."* In this type of usage pattern with *Don't Alert* as the default choice, MyButler served as a sender-powered filter for important and urgent messages.

Their usage pattern exhibits the evidence for the intended benefits of sender-controlled notifications, where senders serve as the filter for unimportant or non-urgent messages, and receivers experience reduced unnecessary interruptions. However, only one out of 11 pairs in the field study belonged to this group and a more extensive study with a larger sample of users is suggested to support the generalizability of our findings.

7.4 Perceived Burdens of Sender-Side Notification Management

Participants had varying opinions on the process of choosing the sender-side notification options. Some felt it was convenient to make the choice with only a single touch, but some felt it was cumbersome especially when the receiver's status indicated (un)availability straightforwardly: *"If the status says 'able to read notifications', MyButler should just send notifications. It was cumbersome having to see the pop-up notification and make a choice"* (F6). F8 also said, *"It'd be nice to have predefined rules like if the status is 'unable to read notifications', then don't alert. For undefined rules, I have to explicitly choose, but if the status says 'available', alert. If 'unavailable', don't alert. Like this."* C2 even preferred to use only the private status sharing feature only and inactivate the notification options: *"If you compare it [the benefit of having sender-side notification options] with its cumbersomeness, I'd prefer to just see the status [without the options]. It appears too often."* Future status sharing and sender-side notification management systems should consider the varying user preferences over features and provide the flexibility to customize which functionality to activate.

8 DISCUSSION

We reflect on the impact of private status sharing and sender-controlled notifications on social awareness and expectations in MIM communications. We then provide design implications based on our study findings.

8.1 Lower Expectation of Immediacy and Social Pressure

In our user study, participants used the status sharing mostly for the Activity statuses that indicate the current activity or task. The interview analysis revealed that showing the current activity or task contributed to a shared understanding of the reason for unavailability. In turn, the shared understanding contributed to relieving the burden and expectation of an immediate response to MIMs. This analysis demonstrates that providing a hint for the *reason* for unavailability could help resolve the expectation of immediacy and social pressure in MIM.

It is also important to consider possible interactions between private status sharing and automated approximations. When they coexist in a system, conflicting status cues can result in diverging consequences, depending on the relationship and trust between users. When the cues are contradictory (e.g., a user is 'active now', but the status says 'busy working'), users with little trust towards each other might rely more on the automated cues and suspect the user of setting a false status. For users who trust each other, using our private sharing could prevent misinterpretations of automated cues and lessen social conflicts by adding more details and context. For example, if Bob is 'active now' to reply to urgent messages but not responding to casual messages, the status 'Bob is busy working' could help the sender understand Bob's situation better.

Design implication: MIMs and other remote communication systems should support users to better express why they cannot immediately respond. Integration of such feature into an existing ecosystem requires careful consideration of how it would interact with existing availability cues.

8.2 Reactive Sharing: Private yet Connected

Reactive sharing preserves privacy by revealing the status message to only people who they *actually chat with.* The message that triggers status sharing also acts as a receipt for who has seen my status when. This reduces the existing concerns of privacy invasion and the feeling of surveillance [2, 5, 16, 37, 48, 52] related to sharing contextual information. Reactive sharing preserves not only privacy but also the personal touch from close relationships. Previous work on sharing contextual information has revealed a case that when shared information replaced direct communication, one partner "missed the personal touch of explicitly asking about each other's availability" and felt "less connected" to each other [16]. While reactive sharing cannot completely replace direct communication because of its nature of sharing the status only in response to a message, our findings suggest that reactive sharing can mitigate the concern of losing direct communication.

Design implication: Reactive sharing could be combined with existing availability indicators (e.g., last seen time, read receipts) or previously studied contextual information (e.g., location [2, 37, 52], motion [5], or combinations of multiple information [16]) to alleviate the concerns of privacy and losing direct communication.

8.3 The Balance Between the Burden of Status Setting vs. Plausible Deniability

The biggest benefit of custom status setting is that the user has full control over the displayed status. The benefit, however, accompanies a tradeoff with the burden of manual settings. In our field study, the perceived burden of status setting varied across participants and some participants utilized the plausible deniability enabled by manual setting as exhibited in three types of butler lies (Section 6.5). The variance in perceived burden and the need for plausible deniability pose an important design question of how to achieve the balance.

One way to reduce the burden without sacrificing plausible deniability is using user-defined automation as described in Section 6.4. Participants suggested creating one-touch shortcuts in the homescreen or on smartwatches; using habitual smartphone actions as triggers for update;

and utilizing calendar items. These techniques reduce the steps involved in manual setting while preserving the user's control over the displayed status. Mobile sensing technologies could also contribute to the automation of status setting. Different contextual information such as motion status [5], location [2], and social setting [45] could be utilized as triggers for a status update. As discussed in Section 6.4, however, it is important to involve the user in status updates, for example by adding a user confirmation step before updating.

In addition, two common types of butler lies in our study were changing the status earlier or later than the actual status change. These behaviors imply that the system could add a user-defined time buffer around the beginning and end of the status.

Design implication: Instead of fully automating status updates for user convenience, status sharing systems should consider the balance for plausible deniability by involving the user in the process of status update (e.g., user-defined rules, user confirmation before an update, or user-defined time buffer around status updates).

8.4 Fine-Grained Sender-Side Notification Control

Sender-controlled notifications were intended to reduce unnecessary interruptions from MIM notifications of unimportant or non-urgent messages. Our field study discovered different usage behaviors of *Alert Now* and *Don't Alert* options, suggesting opportunities for sender-side notification control in MIM systems.

The current design of MyButler provides only binary notification options, *Alert Now* or *Don't Alert*. A message notification is sent either immediately or never. Moreover, the *Don't Alert*-ed messages could go unnoticed until the user opens the app and sees the message. Besides these two options, more fine-grained notification options are possible that do not necessarily alert 'now', e.g., alert after an hour, alert when the status changes, alert when the status becomes available, etc. The more options given to the sender, however, the greater the burden of selection as reported in Section 7.4.

One idea is to support *Smart Alert* that determines the timing of the notification adaptively based on the receiver's status. It is also possible to apply context-aware notification technologies. For example, Park et al. [45] proposed a breakpoint-based smartphone notification deferral in social situations, which defers all notifications until breakpoints, or opportune moments to attend to notifications in social settings. They use mobile sensing techniques to detect the breakpoints such as when a user leaves the table, when a user is alone, and when the conversation pauses for five seconds. By leveraging this approach, if a receiver is 'having a meal' or 'socializing', selecting the *Smart Alert* feature could defer the sender's notifications until proper breakpoints in the context of socializing. Similarly, if a receiver is 'studying', selecting the *Smart Alert* feature could be mapped to delivering the message notifications when the receiver starts moving or using the phone.

Design implication: Our study showed that engaging senders could be effective in sharing the burden of interruption management. An intelligent design of notification deferral options should be explored for wider use of this mechanism.

8.5 Limitations

We acknowledge our field study covers limited demographic diversity as it was conducted in Korea with participants aged from 18 to 29. Previous studies [30, 34, 54, 55] have repeatedly reported heavy use of smartphones and MIM of Korean users. Since the study was conducted on Korean users, the results might not be generalized to different cultures.

Moreover, 12 of 22 participants are undergraduate students who were in their summer break when the study took place. The results therefore do not reflect their behaviors during the academic semesters that are more likely to involve activities that require more concentration and less smartphone distractions, such as taking classes and studying.

In addition, we informed the participants that there will be a reminder notification when there is no change in status for longer than a day. It was intended to remind the participants to update in case they forgot. However, the participants might have updated their status more frequently than they would if it were not a user study, triggered by the notification or to avoid such notifications. This might have affected the quantitative results (e.g., the average number of status setting events, the number of occurrences of each status category).

During the introductory session before the study, we explained that the researchers might see and analyze the participants' status data for the research purpose. This might have affected the status setting behaviors, for example preventing the setting of more intimate or explicit statuses.

We studied the two features, private status sharing and sender-controlled notifications, together in one study. The resulting user behaviors might have been different from studying each feature independently due to interaction effects. For instance, if only private status sharing was enabled, participants might have used fewer statuses that are related to their availability. If only sender-controlled notifications were enabled, senders' decisions might have relied less on the receiver's context and more on the message's importance or value.

In the field study, MyButler was activated only for the 1:1 chat with the partner. The messaging behaviors with private status sharing and sender-controlled notifications remain unknown in the case of having multiple partners and group chats, although group chats are a major part of today's MIM experience [8, 55]. The current MyButler app supports adding more partners and the app's functionality can be scaled seamlessly to group chats for a single partner in the same group chat. For multiple partners in the same group chat, the pop-up notification with each receiver's status and notification options will appear as many times as the number of partners in the chatroom. For future work, the current design should be further tailored to the characteristics of group chats.

9 CONCLUSION

We studied and analyzed how two design concepts for MIM communication, private status sharing and sender-controlled notifications, impact mobile users' chat behavior. We built MyButler, an Android app prototype that enables the application of two design concepts, to work seamlessly with a commercial MIM app, KakaoTalk. We conducted a two-week field study using MyButler followed by retrospective interviews with 11 pairs of friends and couples. Through the field study, we studied how MIM users adopt these two concepts and how the added functions affect their MIM behaviors in their natural, in-situ messaging environments. We found that private status sharing lessens the pressure and expectations for immediate response in MIM, as their personalized and customized status sharing plays the role of shared understanding of real-time (un)availability. Sender-controlled notifications bring the autonomy to the sender to manage their notifications according to the statuses of the receiver's and their own, consideration of receiver-side interruptions, and concerns on message content exposure to others.

ACKNOWLEDGMENTS

This research was supported in part by the Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science and ICT (No.NRF-2017M3C4A7083534), the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No.NRF-2020R1A2C1004062), and the Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No.2016-0-00564, Development of Intelligent Interaction Technology Based on Context Awareness and Human Intention Understanding).

REFERENCES

[1] Daniel Avrahami and Scott E. Hudson. 2006. Responsiveness in Instant Messaging: Predictive Models Supporting Inter-personal Communication. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montreal, Quebec, Canada) *(CHI '06)*. ACM, New York, NY, USA, 731–740. https://doi.org/10.1145/1124772.1124881

[2] Elizabeth Bales, Kevin A. Li, and William Griwsold. 2011. CoupleVIBE: Mobile Implicit Communication to Improve Awareness for (Long-distance) Couples. In *Proceedings of the ACM 2011 Conference on Computer Supported Cooperative Work* (Hangzhou, China) *(CSCW '11)*. ACM, New York, NY, USA, 65–74. https://doi.org/10.1145/1958824.1958835

[3] Louise Barkhuus, Barry Brown, Marek Bell, Scott Sherwood, Malcolm Hall, and Matthew Chalmers. 2008. From Awareness to Repartee: Sharing Location Within Social Groups. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Florence, Italy) *(CHI '08)*. ACM, New York, NY, USA, 497–506. https://doi.org/10.1145/1357054.1357134

[4] James Begole, Nicholas Matsakis, and John Tang. 2004. Lilsys: Sensing Unavailability. *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW*, 511–514. https://doi.org/10.1145/1031607.1031691

[5] Frank R. Bentley and Crysta J. Metcalf. 2007. Sharing Motion Information with Close Family and Friends. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '07)*. ACM, New York, NY, USA, 1361–1370. https://doi.org/10.1145/1240624.1240831

[6] Virginia Braun and Victoria Clarke. 2006. Using thematic analysis in psychology. *Qualitative research in psychology* 3, 2 (2006), 77–101.

[7] Hyunsung Cho, Jinyoung Oh, Juho Kim, and Sung-Ju Lee. 2019. Sender-Controlled Mobile Instant Message Notifications Using Activity Information (Demo). In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services* (Seoul, Republic of Korea) *(MobiSys '19)*. ACM, New York, NY, USA, 669–670. https://doi.org/10.1145/3307334.3329407

[8] Karen Church and Rodrigo de Oliveira. 2013. What's Up with Whatsapp?: Comparing Mobile Instant Messaging Behaviors with Traditional SMS. In *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services* (Munich, Germany) *(MobileHCI '13)*. ACM, New York, NY, USA, 352–361. https://doi.org/10.1145/2493190.2493225

[9] Kakao Corporation. [n.d.]. KakaoTalk. https://play.google.com/store/apps/details?id=com.kakao.talk

[10] Edward S De Guzman, Moushumi Sharmin, and Brian P Bailey. 2007. Should I call now? Understanding what context is considered when deciding whether to initiate remote communication via mobile devices. In *Proceedings of Graphics interface 2007*. ACM, 143–150.

[11] Marc Eisenstadt and Martin Dzbor. 2002. Buddyspace: Enhanced presence management for collaborative learning, working, gaming and beyond. *JabberConf Europe* 89 (2002).

[12] Joel E. Fischer, Chris Greenhalgh, and Steve Benford. 2011. Investigating Episodes of Mobile Phone Activity as Indicators of Opportune Moments to Deliver Notifications. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services* (Stockholm, Sweden) *(MobileHCI '11)*. Association for Computing Machinery, New York, NY, USA, 181–190. https://doi.org/10.1145/2037373.2037402

[13] Joel E. Fischer, Nick Yee, Victoria Bellotti, Nathan Good, Steve Benford, and Chris Greenhalgh. 2010. Effects of Content and Time of Delivery on Receptivity to Mobile Interruptions. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services* (Lisbon, Portugal) *(MobileHCI '10)*. Association for Computing Machinery, New York, NY, USA, 103–112. https://doi.org/10.1145/1851600.1851620

[14] Google. [n.d.]. Firebase Cloud Messaging. https://firebase.google.com/products/cloud-messaging

[15] Google. [n.d.]. Firebase Realtime Database. https://firebase.google.com/products/realtime-database

[16] Carla F. Griggio, Midas Nouwens, Joanna McGrenere, and Wendy E. Mackay. 2019. Augmenting Couples' Communication with Lifelines: Shared Timelines of Mixed Contextual Information. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. ACM, New York, NY, USA, Article 623, 13 pages. https://doi.org/10.1145/3290605.3300853

[17] Jeff Hancock, Jeremy Birnholtz, Natalya Bazarova, Jamie Guillory, Josh Perlin, and Barrett Amos. 2009. Butler Lies: Awareness, Deception and Design. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA) *(CHI '09)*. ACM, New York, NY, USA, 517–526. https://doi.org/10.1145/1518701.1518782

[18] Richard Harper and Stuart Taylor. 2009. Glancephone: An Exploration of Human Expression. In *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services* (Bonn, Germany) *(MobileHCI '09)*. ACM, New York, NY, USA, Article 24, 10 pages. https://doi.org/10.1145/1613858.1613890

[19] Marc Hassenzahl, Stephanie Heidecker, Kai Eckoldt, Sarah Diefenbach, and Uwe Hillmann. 2012. All you need is love: Current strategies of mediating intimate relationships through technology. *ACM Transactions on Computer-Human Interaction* 19, 4 (2012). https://doi.org/10.1145/2395131.2395137

[20] Mariam Hassib, Daniel Buschek, Paweł W. Wozniak, and Florian Alt. 2017. HeartChat: Heart Rate Augmented Mobile Chat to Support Empathy and Awareness. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing

Systems (Denver, Colorado, USA) *(CHI '17)*. ACM, New York, NY, USA, 2239–2251. https://doi.org/10.1145/3025453.3025758

[21] Joyce Ho and Stephen S. Intille. 2005. Using Context-aware Computing to Reduce the Perceived Burden of Interruptions from Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Portland, Oregon, USA) *(CHI '05)*. ACM, New York, NY, USA, 909–918. https://doi.org/10.1145/1054972.1055100

[22] Jin-Hyuk Hong, Sung-Ihk Yang, and Sung-Bae Cho. 2010. ConaMSN: A context-aware messenger using dynamic Bayesian networks with wearable sensors. *Expert Systems with Applications* 37, 6 (2010), 4680–4686.

[23] Roberto Hoyle, Srijita Das, Apu Kapadia, Adam J. Lee, and Kami Vaniea. 2017. Was My Message Read?: Privacy and Signaling on Facebook Messenger. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. ACM, New York, NY, USA, 3838–3842. https://doi.org/10.1145/3025453.3025925

[24] Facebook Inc. [n.d.]. Facebook Messenger. https://play.google.com/store/apps/details?id=com.facebook.orca

[25] Slack Technologies Inc. [n.d.]. Slack. https://play.google.com/store/apps/details?id=com.Slack

[26] WhatsApp Inc. [n.d.]. WhatsApp Messenger. https://play.google.com/store/apps/details?id=com.whatsapp

[27] Instagram. 2019. Threads from Instagram. https://play.google.com/store/apps/details?id=com.instagram.threadsapp

[28] Shamsi T. Iqbal and Brian P. Bailey. 2010. Oasis: A Framework for Linking Notification Delivery to the Perceptual Structure of Goal-directed Tasks. *ACM Trans. Comput.-Hum. Interact.* 17, 4, Article 15 (Dec. 2010), 28 pages. https://doi.org/10.1145/1879831.1879833

[29] Colin Yoon-seung Kang. 2018. KakaoTalk continues to dominate mobile messenger market. http://yna.kr/AEN20180627005100320

[30] Da-jung Kim and Youn-kyung Lim. 2015. Dwelling Places in KakaoTalk: Understanding the Roles and Meanings of Chatrooms in Mobile Instant Messengers. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing* (Vancouver, BC, Canada) *(CSCW '15)*. ACM, New York, NY, USA, 775–784. https://doi.org/10.1145/2675133.2675198

[31] Johannes Knittel, Alireza Sahami Shirazi, Niels Henze, and Albrecht Schmidt. 2013. Utilizing contextual information for mobile communication. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*. ACM, 1371–1376.

[32] Minsam Ko, Seungwoo Choi, Koji Yatani, and Uichin Lee. 2016. Lock n' LoL: Group-Based Limiting Assistance App to Mitigate Smartphone Distractions in Group Activities. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 998–1010. https://doi.org/10.1145/2858036.2858568

[33] Kostadin Kushlev, Jason Proulx, and Elizabeth W. Dunn. 2016. "Silence Your Phones": Smartphone Notifications Increase Inattention and Hyperactivity Symptoms. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. Association for Computing Machinery, New York, NY, USA, 1011–1020. https://doi.org/10.1145/2858036.2858359

[34] Uichin Lee, Joonwon Lee, Minsam Ko, Changhun Lee, Yuhwan Kim, Subin Yang, Koji Yatani, Gahgene Gweon, Kyong-Mee Chung, and Junehwa Song. 2014. Hooked on Smartphones: An Exploratory Study on Smartphone Overuse Among College Students. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. ACM, New York, NY, USA, 2327–2336. https://doi.org/10.1145/2556288.2557366

[35] LlamaLab. 2019. Automate. https://play.google.com/store/apps/details?id=com.llamalab.automate

[36] Danielle Lottridge, Nicolas Masson, and Wendy Mackay. 2009. Sharing Empty Moments: Design for Remote Couples. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Boston, MA, USA) *(CHI '09)*. ACM, New York, NY, USA, 2329–2338. https://doi.org/10.1145/1518701.1519058

[37] Clara Mancini, Yvonne Rogers, Keerthi Thomas, Adam N. Joinson, Blaine A. Price, Arosha K. Bandara, Lukasz Jedrzejczyk, and Bashar Nuseibeh. 2011. In the best families: Tracking and relationships. *Conference on Human Factors in Computing Systems - Proceedings* (2011), 2419–2428. https://doi.org/10.1145/1978942.1979296

[38] Abhinav Mehrotra, Robert Hendley, and Mirco Musolesi. 2016. PrefMiner: Mining User's Preferences for Intelligent Mobile Notification Management. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Heidelberg, Germany) *(UbiComp '16)*. ACM, New York, NY, USA, 1223–1234. https://doi.org/10.1145/2971648.2971747

[39] Abhinav Mehrotra, Mirco Musolesi, Robert Hendley, and Veljko Pejovic. 2015. Designing Content-driven Intelligent Notification Mechanisms for Mobile Applications. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Osaka, Japan) *(UbiComp '15)*. ACM, New York, NY, USA, 813–824. https://doi.org/10.1145/2750858.2807544

[40] Abhinav Mehrotra, Veljko Pejovic, Jo Vermeulen, Robert Hendley, and Mirco Musolesi. 2016. My Phone and Me: Understanding People's Receptivity to Mobile Notifications. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '16)*. ACM, New York, NY, USA, 1021–1032. https://doi.org/10.1145/2858036.2858566

[41] Tadashi Okoshi, Jin Nakazawa, and Hideyuki Tokuda. 2014. Attelia: Sensing User's Attention Status on Smart Phones. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication* (Seattle, Washington) *(UbiComp '14 Adjunct)*. ACM, New York, NY, USA, 139–142. https://doi.org/10.1145/2638728.2638802

[42] Tadashi Okoshi, Hiroki Nozaki, Jin Nakazawa, Hideyuki Tokuda, Julian Ramos, and Anind K. Dey. 2016. Towards Attention-Aware Adaptive Notification on Smart Phones. *Pervasive Mob. Comput.* 26, C (Feb. 2016), 17–34. https://doi.org/10.1016/j.pmcj.2015.10.004

[43] Tadashi Okoshi, Julian Ramos, Hiroki Nozaki, Jin Nakazawa, Anind K. Dey, and Hideyuki Tokuda. 2015. Reducing Users' Perceived Mental Effort Due to Interruptive Notifications in Multi-device Mobile Environments. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Osaka, Japan) *(UbiComp '15)*. ACM, New York, NY, USA, 475–486. https://doi.org/10.1145/2750858.2807517

[44] Antti Oulasvirta. 2008. Designing Mobile Awareness Cues. In *Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services* (Amsterdam, The Netherlands) *(MobileHCI '08)*. ACM, New York, NY, USA, 43–52. https://doi.org/10.1145/1409240.1409246

[45] Chunjong Park, Junsung Lim, Juho Kim, Sung-Ju Lee, and Dongman Lee. 2017. Don'T Bother Me. I'm Socializing!: A Breakpoint-Based Smartphone Notification System. In *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing* (Portland, Oregon, USA) *(CSCW '17)*. ACM, New York, NY, USA, 541–554. https://doi.org/10.1145/2998181.2998189

[46] Elin Rønby Pedersen. 2001. Calls.Calm: Enabling Caller and Callee to Collaborate. In *CHI '01 Extended Abstracts on Human Factors in Computing Systems* (Seattle, Washington) *(CHI EA '01)*. ACM, New York, NY, USA, 235–236. https://doi.org/10.1145/634067.634207

[47] Veljko Pejovic and Mirco Musolesi. 2014. InterruptMe: Designing Intelligent Prompting Mechanisms for Pervasive Applications. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (Seattle, Washington) *(UbiComp '14)*. ACM, New York, NY, USA, 897–908. https://doi.org/10.1145/2632048.2632062

[48] Martin Pielot, Rodrigo de Oliveira, Haewoon Kwak, and Nuria Oliver. 2014. Didn't You See My Message?: Predicting Attentiveness to Mobile Instant Messages. In *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems* (Toronto, Ontario, Canada) *(CHI '14)*. ACM, New York, NY, USA, 3319–3328. https://doi.org/10.1145/2556288.2556973

[49] Anand Ranganathan, Roy H Campbell, Arathi Ravi, and Anupama Mahajan. 2002. Conchat: A context-aware chat program. *IEEE Pervasive computing* 1, 3 (2002), 51–57.

[50] Yasmeen Rashidi, Kami Vaniea, and Jean L. Camp. 2016. Understanding Saudis' privacy concerns when using WhatsApp. In *Usable Security and Privacy (USEC) 2016*.

[51] Lindsay Reynolds, Madeline E. Smith, Jeremy P. Birnholtz, and Jeff T. Hancock. 2013. Butler Lies from Both Sides: Actions and Perceptions of Unavailability Management in Texting. In *Proceedings of the 2013 Conference on Computer Supported Cooperative Work* (San Antonio, Texas, USA) *(CSCW '13)*. ACM, New York, NY, USA, 769–778. https://doi.org/10.1145/2441776.2441862

[52] Emily Schildt, Martin Leinfors, and Louise Barkhuus. 2016. Communication, Coordination and Awareness Around Continuous Location Sharing. In *Proceedings of the 19th International Conference on Supporting Group Work* (Sanibel Island, Florida, USA) *(GROUP '16)*. ACM, New York, NY, USA, 257–265. https://doi.org/10.1145/2957276.2957289

[53] Albrecht Schmidt, Antti Takaluoma, and Jani Mäntyjärvi. 2000. Context-Aware Telephony Over WAP. *Personal Ubiquitous Comput.* 4, 4 (Jan. 2000), 225–229. https://doi.org/10.1007/s007790070008

[54] In-geon Shin, Jin-min Seok, and Youn-kyung Lim. 2018. Too Close and Crowded: Understanding Stress on Mobile Instant Messengers Based on Proxemics. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. ACM, New York, NY, USA, Article 615, 12 pages. https://doi.org/10.1145/3173574.3174189

[55] In-geon Shin, Jin-min Seok, and Youn-kyung Lim. 2019. Ten-Minute Silence: A New Notification UX of Mobile Instant Messenger. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (Glasgow, Scotland Uk) *(CHI '19)*. ACM, New York, NY, USA, Article 442, 13 pages. https://doi.org/10.1145/3290605.3300672

[56] D. Siewiorek, A. Smailagic, J. Furukawa, A. Krause, N. Moraveji, K. Reiger, J. Shaffer, and Fei Lung Wong. 2003. SenSay: a context-aware mobile phone. In *Seventh IEEE International Symposium on Wearable Computers, 2003. Proceedings.* 248–249. https://doi.org/10.1109/ISWC.2003.1241422

[57] Skype. [n.d.]. Skype. https://play.google.com/store/apps/details?id=com.skype.raider

[58] Cary Stothart, Ainsley Mitchum, and Courtney Yehnert. 2015. The Attentional Cost of Receiving a Cell Phone Notification. *Journal of experimental psychology. Human perception and performance* (06 2015). https://doi.org/10.1037/xhp0000100

[59] John C. Tang, Nicole Yankelovich, James Begole, Max Van Kleek, Francis Li, and Janak Bhalodia. 2001. ConNexus to Awarenex: Extending Awareness to Mobile Users. In *Proceedings of the SIGCHI Conference on Human Factors in*

Computing Systems (Seattle, Washington, USA) *(CHI '01)*. ACM, New York, NY, USA, 221–228. https://doi.org/10.1145/365024.365105

[60] Alexander Thayer, Matthew J. Bietz, Katie Derthick, and Charlotte P. Lee. 2012. I love you, let's share calendars: Calendar sharing as relationship work. *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW* (2012), 749–758. https://doi.org/10.1145/2145204.2145317

[61] Alexander Thayer, Behzod Sirjani, and Charlotte P. Lee. 2013. Recalibrating the ratio: Enacting accountability in intimate relationships using shared calendars. *Proceedings of the ACM Conference on Computer Supported Cooperative Work, CSCW* (2013), 203–213. https://doi.org/10.1145/2441776.2441801

[62] Dominik Weber, Alexandra Voit, and Niels Henze. 2018. Notification Log: An Open-Source Framework for Notification Research on Mobile Devices. In *Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers* (Singapore, Singapore) *(UbiComp '18)*. ACM, New York, NY, USA, 1271–1278. https://doi.org/10.1145/3267305.3274118

[63] Lance Whitney. 2018. *'Do Not Disturb While Driving' on iOS*. Ziff Davis, LLC. PCMag Digital Group. https://www.pcmag.com/feature/359890/how-to-turn-on-do-not-disturb-while-driving-on-ios

[64] Zhang Xiaolei. 2009. Towards pervasive instant messaging and presence awareness. 5, 1 (01 Jan 2009), 42–60. https://doi.org/10.1108/17427370910950302