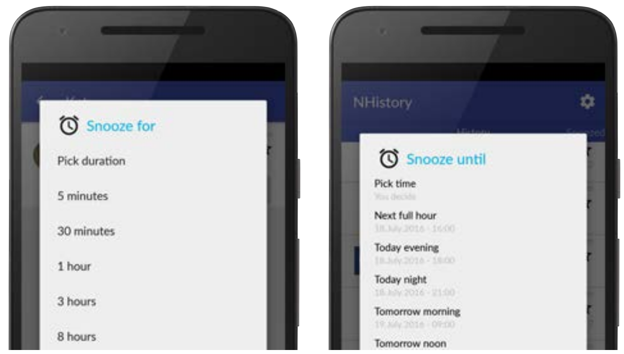
Snooze! Investigating the User-Defined Deferral of Mobile Notifications

Dominik Weber[1], **Alexandra Voit**[1], **Jonas Auda**[2], **Stefan Schneegass**[2], **Niels Henze**[1]
[1]VIS, University of Stuttgart
Stuttgart, Germany
{firstname.lastname}@vis.uni-stuttgart.de
[2]University of Duisburg-Essen
Essen, Germany
{firstname.lastname}@uni-due.de

ABSTRACT
Notifications on mobile devices are a prominent source of interruptions. Previous work suggests using opportune moments to deliver notifications to reduce negative effects. In this paper, we instead explore the manual deferral of notifications. We developed an Android app that allows users to "snooze" mobile notifications for a user-defined amount of time or to a user-defined point in time. Using this app, we conducted a year-long in-the-wild study with 295 active users. To complement the findings, we recruited 16 further participants who used the app for one week and subsequently interviewed them. In both studies, snoozing was mainly used to defer notifications related to people and events. The reasons for deferral were manifold, from not being able to attend notifications immediately to not wanting to. Daily routines played an important role in the deferral of notifications. Most notifications were deferred to the same day or next morning, and a deferral of more than two days was an exception. Based on our findings, we derive design implications that can inform the design of future smart notification systems.

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI): Miscellaneous

Author Keywords
Mobile notifications; interruptions; mobile HCI; notification management; deferral; in-the-wild; in-situ.



(a) Duration Method (b) Point-in-Time Method

Figure 1. The available snooze options depend on the selected method. Left: Duration. Right: Point-in-time (depending on the time of the day).

INTRODUCTION
Notifications are a key feature of current smartphones. They are used to proactively inform users about incoming messages, emails as well as social network updates and events [30]. Notifications use visual cues, auditory signals, and haptic feedback to gain the user's attention [14]. An increasing number of mobile applications issue notifications for various purposes [34]. As a result, users of mobile devices currently receive a large number of notifications on a daily basis [30, 34]. The recipients of notifications, especially on mobile devices which accompany its users throughout the day, are coping with a mostly uncontrolled stream of proactively delivered information. Suitably controlling this stream is a major research challenge that is under heavy investigation nowadays.

Notifications can induce negative effects. They can be distracting, might cause negative emotions, or are just not important for their recipients [26, 30, 32, 34]. One strand of research aims to reduce distraction by finding more suitable moments for notification delivery. With *SMS* and instant messaging (*IM*) services generating numerous notifications, further research is needed to develop more appropriate notification management services [4]. Previous work found that delaying the delivery of notifications to opportune moments helps to reduce distraction [10, 22, 23, 24, 26, 36]. Contextual information about the user can be used to determine suitable moments to deliver notifications. These moments promise to alleviate negative effects on the recipient of notifications.

Despite the increasing interest in finding opportune moments for delivering notifications, little is known about the type of notification that should be delayed and how long users like the delay to be. Therefore, we developed the Android application *NHistory* that enables users to *snooze* notifications similar to snoozing an alarm clock. In contrast to previous approaches, the app lets the recipient of a notification specify the moment when the notification should be redelivered. Notifications can be snoozed for a specific duration or to a later point-in-time, and *NHistory* reissues them after the specified period. Users

MobileHCI '18, September 3–6, 2018, Barcelona, Spain

ISBN 978-1-4503-5898-9/18/09...$15.00
DOI: https://doi.org/10.1145/3229434.3229436

can also access a history of all notifications they have received, enabling them to go through the notifications and snooze desired notifications to more appropriate moments. By snoozing notifications to more suitable times, users can reduce the number of pending notifications in the notification drawer. While this is indeed a positive aspect of the app, we want to note that this does not necessarily mitigate adverse effects on the notifications' recipient. In some cases, it might be even worse than an uncontrolled notification delivery, because a snoozed and redelivered notification may induce a second interruption. However, observing how a recipient of a notification postpones its arrival enables us to gain insights into the types of notifications users want to defer and how long users would like the deferral to be.

In this paper, we report a year-long in-the-wild study with 295 active users. The goal of the study was to understand which notifications users want to defer and the periods of time users want to defer them. We focused on deriving insights into how users postpone notifications rather than eliminating negative effects that might influence the notification recipient directly. We collected usage data including applications that issued notifications, how long notifications were snoozed for, and when they were re-triggered. To complement the collected quantitative data, we conducted a second study with 16 participants for one week and subsequently interviewed them. In both studies, snoozing was mainly used to defer notifications related to people and events. Reasons for the deferral of notifications were manifold and aligned with daily routines. Most notifications were deferred to the same day or the next morning, indicating an upper bound for the deferral of notifications. Based on our findings, we derive design implications that can inform the design of future smart notification systems.

RELATED WORK

Notifications received on mobile devices are used to inform the user about incoming messages, emails, social network updates, and events [30, 39]. Using notifications, these devices compete for the attention of their user [12, 19]. Besides the informative value of notifications, the delivery can cause negative effects such as interruption of the user. In the following, we present previous work on notifications and interruptions as well as opportune moments for presenting notifications to users.

Notifications and Interruptions

The number of generated notifications is constantly increasing [30]. Smartphone users themselves often do not realize how many notifications they receive [40]. Sahami et al. found that a large number of notifications is issued by messaging applications [34, 38]. On the one hand, users value notifications issued by such applications. On the other hand, not all notifications that users receive are considered important. Church and Oliveira compared *SMS* to instant messaging applications like *WhatsApp* [4]. Their study revealed several concerns regarding *WhatsApp* messages and notifications, e.g., coping with too many messages or interruptions, and the fear of missing business-related messages if notification modalities are switched off. Understanding how users handle messaging notifications might help to build messaging services which do not overload users by issuing too many notifications.

A notification recipient might benefit from valuable information that he or she receives in a proactive manner [7, 19]. However, previous studies show that notifications can cause negative effects to its recipient. Ill-timed notifications can also distract or interrupt the recipient [16, 30]. Further negative effects include decreased productivity and slower and more error-prone performance [1, 3, 11, 28, 35].

Merely disabling notifications is not a suitable solution to alleviate corresponding adverse effects on the recipient of a notification [32, 33]. The desire to meet social expectations or the fear to appear rude by not replying to messages on time are reasons for keeping notifications enabled. Further, notifications act as reminders for important events. Stawarz et al. investigated efficient reminders for taking medicine [37]. Time-based reminders offer only a small benefit to users because often the reminder cannot be postponed. Immediate action is required by the user so that taking medicine is not forgotten. Stawarz et al. propose that users should be able to defer notifications issued by reminders if they are not able to respond immediately. To reduce the negative effects of the increasing number of notifications, previous research focused on developing models, rules, and systems that manage notification delivery. However, users might not accept a notification management systems that removes important notifications [19].

Opportune Moments

To identify moments in which notifications can be best presented to the user, researchers exploited contextual information, notification content, and personal traits. Corno et al. introduced a smart notification system that uses machine learning to manage incoming notifications [6]. These algorithms use information about the context of the user and the user itself such as location and user's activity. Several approaches used sensor data to develop context-aware notification systems [12, 15]. Kern and Schiele introduced a model that makes use of body-worn sensor data. The model supports notification classification according to the user's current context to determine his or her interruptibility. Horvitz et al. introduced *BusyBody*, a system running on a desktop computer that predicts the cost of an interruption on a user [13]. It analyzes desktop events and the user's context (e.g., if he or she is speaking or not) to train a model. Mehrotra et al. used the context of a notification recipient in combination with the content of the notification to realize a non-disruptive notification mechanism [18]. Notifications were grouped into different categories according to the application that issued the notification as well as the relationship between sender and receiver. They showed that their machine learning based predictors could outperform user-defined rules for notification delivery on smartphones. Further, Mehrotra et al. proposed an interruptibility management solution for mobile notifications [17]. Their system extracts rules to handle notifications automatically based on how the user is interacting with his or her device.

Yuan et al. proposed a model which can determine the interruptibility of its user [41]. The model determines if the user is available to react to an incoming notification using sensory data and additional information such as the mood of the user. They found that including personality traits is important for predicting interruptibility. Siewiorek et al. proposed *SenSay*, a context-aware mobile phone [36]. It uses sensors to acquire contextual information about its user. Using this information the device can switch to the uninterruptible state. In this state, unwanted interruptions will not be delivered to the user.

Pielot et al. investigated if mobile phone usage patterns can be utilized to train a machine learning model that is able to detect boredom [31]. They suggest that bored people may be more appreciative of incoming notifications. Dingler et al. investigated if detected boredom can be used to engage a user in micro-learning sessions through notifications [9]. Indeed people search for stimulation during boredom, but they found that a mentally demanding task is not suitable. To voluntary engage users to interact with recommended content Pielot et al. used a machine learning approach to determine opportune moments for notification delivery [29]. Higher user engagement was observed if notifications were issued at these moments. Further, they suggest observing past interest in notification content to reduce future interruptions.

To further mitigate negative effects caused by interruptions previous research investigated the suitability of breakpoints for notification delivery. A breakpoint promises to be more suitable for notification delivery because they occur between two consecutive activities [21]. Recipients of interruptions during these moments may not be as strongly influenced by interruption related adverse effects as they would be while performing a specific activity or task. Fischer et al. reported that people attended to notifications quicker at the end of a mobile interaction (i.e., calling a contact or reading *SMS*) compared to notifications received at random times [10]. Different tools were suggested to cope with the number of incoming notifications. Ho and Intille used data from accelerometers to automatically detect transitions between physical activities [12]. They suggested that these transitions might be suitable for reducing negative effects caused by interruption from mobile devices.

Okoshi et al. proposed *Attelia*, a service that identifies breakpoints for notification delivery [22, 23, 24, 25]. *Attelia* runs on the user's smartphone and can detect breakpoints of the user's activity on his or her mobile device based on running applications and machine learning techniques. Further, it can detect physical breakpoints through smartwatches. Investigating breakpoints in the wild, Okoshi et al. conducted a large-scale study with their breakpoint detection system included into a popular Android application. They reduced the response time to the delayed notifications significantly. They also observed a continuously increasing number of clicks as well as a higher level of user engagement. Park et al. proposed the breakpoint-based *Social Context-Aware smartphone Notification system (SCAN)* [26]. *SCAN* can identify breakpoints to which it defers incoming notifications. Park et al. reported that *SCAN* could help the participants to better focus on their social interaction. Pejovic and Musolesi proposed *InterruptMe*, a library for interruption management for Android [27]. With the use of contextual information about the user, the library determines whether he or she is interruptible. *InterruptMe* can inform other apps when an opportune moment for an interruption occurs.

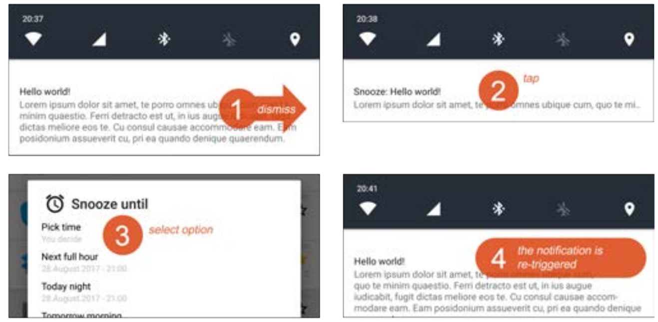


Figure 2. Snoozing a notification via the notification drawer. 1. User dismisses the notification. 2. *NHistory* briefly shows a "snooze notification". 3. Tapping on the snooze notification opens the snooze options. 4. After the selected duration, the notification is re-triggered.

In summary, previous work showed that delaying the delivery of notifications to opportune moments can partially reduce notifications' negative effects. A body of work focused on automatically finding opportune moments for notification delivery. However, little is known about which types of notifications and to what times users would defer notifications manually.

SYSTEM

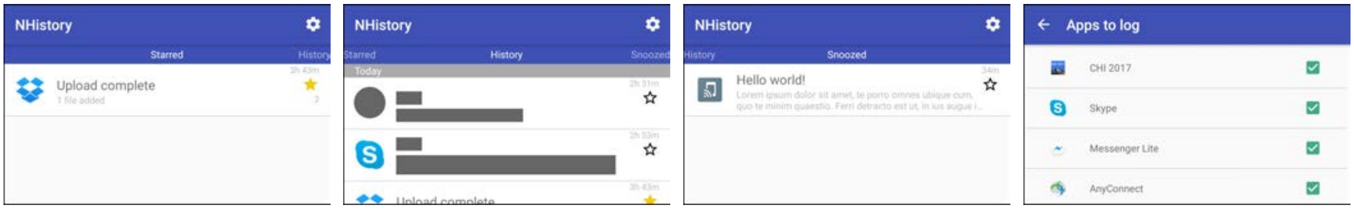
In contrast to previous work, we investigate the manual deferral of notifications to study reasons for deferring notifications, selected moments, and types of notifications to better understand why and how users defer notifications. Therefore, we developed an Android application that enables users to defer notifications manually. In the following, we first summarize Android's notification mechanisms and afterward describe how *NHistory* extends the Android system.

Notifications in Android

Notifications play a central role in the Android mobile operating system. They consist, at the minimum, of a small icon and two lines of text. Notifications are accessible from the notification drawer; a list of active notifications that can be accessed by swiping down from the top of the screen. Android's status bar displays icons to indicate active notifications in the notification drawer. Newer versions of Android display notifications on the lock screen as well. Notifications are ephemeral. Users can dismiss single or all active notifications at once. Dismissing a notification removes it from the system, an action that cannot be undone. An active notification might be seen as a "nagging reminder" to take action while dismissing a notification might cause the user to forget about taking said action. Users have to decide if they want to respond to the notification (e.g., reply to an instant message), dismiss the notification, or just ignore it until a later point in time [5].

NHistory

We developed *NHistory* to investigate the user-defined deferral of mobile notifications. The app provides a timeline of all active and dismissed notifications, sorted by the time of creation. Further, the app allows users to dismiss notifications and automatically re-triggers them after a user-defined duration or at a user-defined point-in-time. This behavior of temporarily muting is known from alarm clocks and generally referred to as



(a) Starred Notifications (b) Notification History (c) Snoozed Notifications (d) App Blacklist

Figure 3. Screenshots of the *NHistory* app. From left to right: List of starred notifications, notification history showing active and past notifications (main view), list of snoozed notifications, and blacklist to exclude notifications from specific apps from the history.

"snoozing" an alarm. To enable this functionality, the app uses Android's *Notification Listener* API [8]. When first opening *NHistory*, users are informed about its data collection. To use the app, users have to agree to the data collection explicitly; declining closes the app. After accepting the data collection, the user is prompted to grant *NHistory* the permission to access notifications. Then, the app's main view is shown (Figure 3b). The *history* view shows a list of active and past notifications. Users can tap on notifications to expand them. "Ongoing" notifications, like downloads, are ignored by *NHistory*. Clicking on the star icon next to a notification allows users to bookmark them. Starred notifications are shown to the left of the history (Figure 3a) and can be removed by clicking on the star icon again. To the right of the history, notifications that are currently snoozed are listed (Figure 3c). In the settings, users can set a limit of how many notifications should be stored in the history or exclude specific apps from appearing in the history using a blacklist (Figure 3d). Users can access *NHistory* by either clicking on its icon in the launcher or through an optional persistent notification in the notification drawer.

Snoozing Notifications

NHistory offers multiple ways to snooze notifications. In the app itself, users can long-press on a notification or expand a notification and subsequently click on a snooze button. Active notifications are automatically dismissed from the notification drawer when snoozing them. While the Android system does not allow modifying the notification drawer, we also implemented an option to snooze notifications from the drawer directly. *NHistory* detects when a user dismisses a notification from the notification drawer. The app then creates a temporary notification in its place, that is shown for five seconds. Clicking this temporary "snooze notification" allows the user to trigger the snooze action for the dismissed notification (see Figure 2). The user is then prompted to define *how long* or *until when* the notification should be snoozed. We implemented the two "snooze" methods *duration* and *point-in-time* with corresponding options. Both methods provide a set of eight options. The *duration* method features fixed options, while the *point-in-time* options depend on the time of the day (see Figure 1). Both methods feature a custom duration or custom point-in-time as the first option. On the first start of the app, one of the snooze methods is randomly assigned. Users can change the assigned method in the settings.

Data Collection

NHistory collects information about the device, notification meta-data, and interaction with the app. Upon installation of *NHistory*, a random identifier is created. All requests to our server use a secure connection and include the identifier. Device information is sent periodically, including device type, Android version, and system language. The app records all major user interaction including starring and snoozing notifications, and blacklisting apps from the history. Further, package names and timestamps of all notifications created on the device are recorded. We did not record any text or information that could be used to identify users. All recorded data is queued for sending until a Wi-Fi connection is established.

IN-THE-WILD STUDY

To gain insights into the types of notifications users want to defer, and for how long, we conducted an in-the-wild study.

Method and Participants

We released *NHistory* on the Google Play Store as a free download. People across the globe were able to download and use the application. The app automatically reported anonymized usage and notification meta-data back to us. Users who downloaded the app were informed about the data collection on the first start of the app. All users had to consent to the data collection to use the app.

Between January 12, 2017, and January 12, 2018, *NHistory* was installed on 1,555 devices. According to the Google Play Store statistics, the app was downloaded from 95 countries, with most downloads originating from India (43.08%), the United States (12.43%), and Germany (4.25%). The most popular device languages were variants of English (81.28%), followed by German (3.67%). Android versions 6.0 (33.89%), 7.0 (27.65%), and 7.1 (16.98%) contributed the most to the user base. Only 13 users installed the app on Android tablets; the remaining 1,542 installs were on smartphones. Thirty users rated the app, resulting in an average rating of 4.37 stars (1=worst; 5=best).

Results

Of the 1,555 users who downloaded the app, over half (876) agreed to the terms of the study on the first start. It is important to note that users tend to try out free apps and quickly uninstall them if they do not fulfill their expectations. Indeed, a number of users uninstalled the app right after the setup. 581 users used the app for less than two days. Thus, we excluded them from our analysis. The remaining 295 users used the app between 2 and 360 days (M=46; SD=68; Md=15 days).

	In-The-Wild Study				Controlled Study			
Category	**Normalized%**	**Total Events**	**Users**	**Apps**	**Normalized%**	**Total Events**	**Users**	**Apps**
Calendar/Reminder	18.68	430	41	26	4.47	5	3	3
Email	8.19	111	30	11	13.66	21	5	5
Game	2.95	22	6	7	3.33	1	1	1
Health/Fitness	.25	15	5	5	2.76	3	2	2
Media	3.08	26	12	9	.35	1	1	1
News	1.82	8	5	6	4.39	4	2	3
Phone	8.45	102	22	10	4.44	2	2	1
Shopping/Finance	3.75	26	16	13	2.22	1	1	1
SMS/IM	28.23	251	64	30	49.15	48	13	6
Social	9.20	73	28	20	4.11	9	3	4
System	6.56	40	24	12	8.03	5	4	2
Tool	8.83	87	27	36	3.07	9	2	4

Table 1. Normalized distribution of the initial snooze events for all users and all categories. Left: Values for the in-the-wild study (1,191 events). Right: Values for the controlled study (109 events). In both studies, notifications of the category *SMS/IM* were proportionally snoozed most often.

Notifications and Apps

In total, we logged 20,345,277 notifications from 3,667 apps. Users had on average 44 apps notifying them (SD=27). We recorded on average 1,960 notification events per user per day (SD=3,942; Md=832). At first glance, these numbers seem unusually high. The reason for these numbers is that in Android updating an existing notification is realized by replacing the original notification. Some apps continuously update notifications in the background, e.g., to display location updates or battery statistics. Although these background updates might happen every other second, they are often not noticeable to the user because they happen silently. Since we did not record the text of notifications, we were unable to filter these updates. The apps that created the most notifications were *Google Maps* (2,049,889 notifications; 210 users), *Power Clean* (1,665,971 notifications; 6 users), and *WhatsApp* (1,325,423 notifications; 165 users).

Most notifications were created between 8pm and 10pm, peaking at 9pm. At 4am the least number of notifications were created. Regarding the days of the week, the notifications were evenly distributed over weekdays and the weekend.

Notification History Blacklist

109 of 295 users made use of the option to exclude apps from the history. These users excluded between 1 and 435 apps (M=46; SD=81; Md=9), totaling in 2,760 different apps. Some users added apps to the blacklist that did not yet post notifications. The apps excluded by most users were the *Android OS* (44 users), the *Android System UI* (43 users), and the *Google Play Store* (42 users).

Starred Notifications

Only 52 of 295 users starred notifications. Users starred between 1 and 48 notifications (M=3; SD=7; Md=1) for at least one hour, totaling in 159 star events from 58 apps. Notifications from apps that were starred by more than one user include *Google Calendar* (32 events; 4 users), *Facebook* (7 events; 5 users), and *Google Keep* (6 events; 4 users).

Snoozed Notifications

We recorded 2,648 snooze events from 151 users and 219 apps. Since users interested in the snooze functionality would likely try it after installing the app, we excluded snooze events from the day when *NHistory* was installed, resulting in 2,390 snooze events from 129 users and 185 apps. 66 users were randomly assigned the *point-in-time (pit)* method during installation, and 63 users were assigned the *duration* method. 79 users stayed with their assigned method (42 *duration*, 37 *pit*), 32 users switched to the other method (14 from *duration* to *pit*, 18 from *pit* to *duration*), and 18 users switched but returned to their assigned method (7 *duration*, 11 *pit*).

Snoozing of \ Compared to	Health/Fitness			SMS/IM		
	T	diff.	p	T	diff.	p
Calendar/Reminder	5.00	↗	< .001*	40.61		= .002
Email	27.00	↗	< .001*	51.50	↘	< .001*
Game	3.20		= .002	39.50	↘	< .001*
Health/Fitness	—		—	9.50	↘	< .001*
Media	2.67		= .001	38.00	↘	< .001*
News	2.50		= .002	25.50	↘	< .001*
Phone	17.50	↗	< .001*	64.00	↘	< .001*
Shopping/Finance	3.50		= .001	65.00	↘	< .001*
SMS/IM	9.50	↗	< .001*	—		—
Social	7.00	↗	< .001*	73.50	↘	< .001*
System	20.00	↗	< .001*	61.00	↘	< .001*
Tool	7.00	↗	< .001*	34.47		= .001

Table 2. Results of the post-hoc analysis with Wilcoxon signed-rank tests of the normalized snooze categories (in-the-wild study). p-values are Holm-Bonferroni adjusted. Significant differences are marked with "*", significant less snoozed categories with "↘", and significantly more snoozed categories with "↗".

Looking closer at the source of the 2,390 snooze events showed that 1,191 individual notifications were snoozed. 728 notifications were snoozed once, and 463 notifications were snoozed multiple times, with a single notification being snoozed 18 times by a user. We will now report on the 1,191 initial snooze events, followed by the re-snooze events. The 1,191 initial snooze events by 129 users break down to 1 to 402 snooze events per user (M=9; SD=36; Md=3). The apps that were snoozed the most were the *Google Calendar* (340 events; 18 users), *WhatsApp* (72 events; 21 users), and *Outlook* (58 events; 4 users).

To abstract from single apps, two researchers independently categorized the 185 apps. The resulting 12 categories are based on Google Play Store listings and prior literature [34] (see Table 1). Disagreements were discussed until an agreement was reached. The left part of Table 1 shows the normalized distribu-

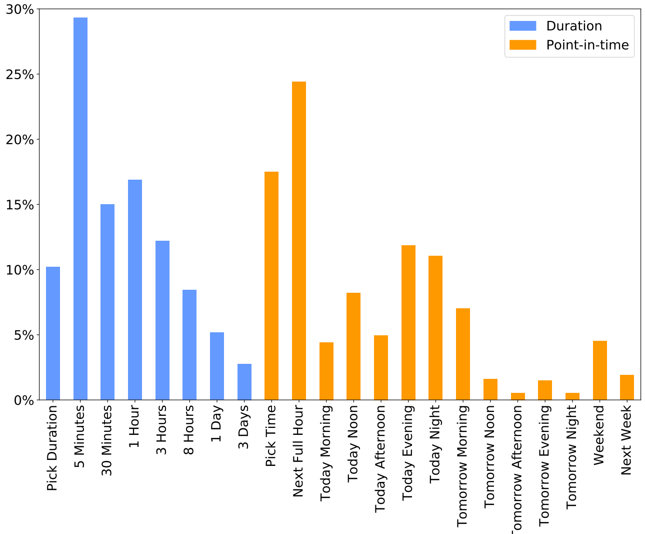


Figure 4. Normalized distribution of selected options for all 1,191 initial snooze events, both methods, and all users (in-the-wild study). Duration: 797 events from 75 users. Point-in-time: 394 events from 64 users.

tion of snooze events based on the categories. The categories with the most snooze events were *SMS/IM* (28.23%), *Calendar/Reminder* (18.68%), and *Social* (9.20%). We conducted a Friedman test to compare the normalized data of snoozed notifications from different categories. The results show that the category has an influence on the users' snoozing behavior ($\chi^2(11) = 189.96, p < .001$).

We conducted a post-hoc analysis with Wilcoxon signed-rank tests and applied a Holm-Bonferroni correction (see Table 2). The results show that notifications from the category *Health/Fitness* were significantly *less often* snoozed than from the categories *SMS/IM*, *Calendar/Reminder*, *Social*, *Tool*, *Phone*, *Email*, and *System*. Further, notifications from the category *SMS/IM* were significantly *more often* snoozed than all other categories except *Calendar/Reminder* and *Tool*.

The usage of the *duration* and *point-in-time* methods was equally distributed. Normalized over all users, 54% of the 1,191 initial snooze events were executed using the *duration* method and 46% using *point-in-time*. We normalized the selected options for all users and both methods (see Figure 4). The most popular options for the *duration* method were "5 minutes" (29.32%), "1 hour" (16.88%), and "30 minutes" (14.99%). The custom duration option was mainly used to fill the gaps between the pre-defined options, with most durations being between 10-15 minutes and 2 hours. The median custom duration was 1 hour and the maximum duration 5 days. For the *point-in-time* method, the most popular options were "next full hour" (24.42%), picking a custom time (17.50%), and "today evening" (11.87%). For custom points-in-time, users mostly selected times dividable by 15 minutes, with a median snooze time of 5.62 hours, and a maximum point-in-time of 4.67 days in the future.

Figure 5 shows the posted, snoozed, and re-triggered times of the initial snooze events. Most notifications that were snoozed were posted in the morning and early noon. Users quickly

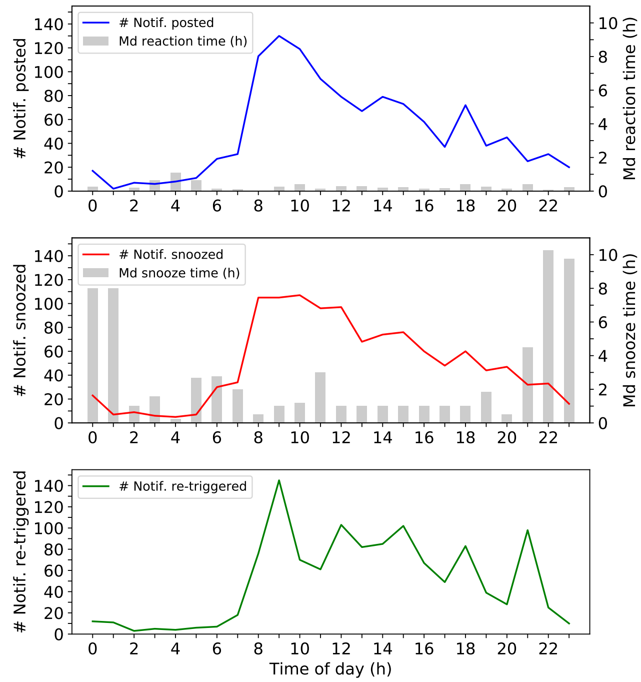


Figure 5. Plots showing the distribution of the 1,191 initial snooze events from the in-the-wild study. The line charts show the time distribution when the notifications were posted (top), snoozed (middle), and re-triggered (bottom). The bar charts show the median reaction time (top) and median snooze time (middle).

attended the notifications, resulting in low reaction times (Md=35*min*). Consequently, the distribution of the snooze events follows the posted events, peaking in the morning and dropping over the course of the day. During the day we observed notifications being snoozed for shorter time spans. In the evening and until after midnight the time spans notifications were snoozed for increased by several hours, as users snoozed the notifications until the next morning. Most snoozed notifications were re-triggered during the day, between 8am and 10pm. We can see peaks at 9am, 12pm, 3pm, 6pm, and 9pm. 9am being the highest peak can be explained by notifications that were snoozed for a short time in the early morning and notifications from the previous day. These times are likely influenced by the pre-defined options of *NHistory* but might also correlate with before and after work hours, and lunch breaks. We want to highlight that 79.26% of the notifications were posted, snoozed and re-triggered on the same day. Only 16.54% were re-triggered on the following day and 4.20% over multiple days. This results in an overall mean deferral time of 689 minutes (SD=1,543; Md=274), between a notification being posted and eventually re-triggered.

As mentioned earlier, 463 notifications were snoozed more than once (Min=2; Max=18; Md=3). These notifications were mainly from the categories *Calendar/Reminder* (55.94%), *SMS/IM* (15.77%), and *Email* (10.15%). Looking at the times when these notifications were initially posted and finally re-triggered, we still found that most were on the same day (69.61%), some on the following day (20.49%), and more than two days being an exception (9.91%).

Summary
Temporarily snoozing was more popular than permanently starring notifications, indicating the ephemeral nature of notifications. We assume that users only snoozed particularly important notifications that they were unable to attend directly. Users mainly snoozed notifications from the categories *SMS/IM* and *Calendar/Reminder*. They were fast to snooze notifications, with overall low reaction times. The number of snooze events spiked in the morning and declined over the course of the day. Most notifications were re-triggered in the morning, at noon and in the evening. These times likely correlate with before and after work hours, and lunch breaks. We found that users snoozed notifications mainly to the same day. Only few notifications were snoozed to the following day and even less to the day after.

CONTROLLED STUDY
The in-the-wild study provided insights into the types of notifications and times users are interested in deferring notifications. However, we know little about the users' motivation. To complement the quantitative results, we conducted a second in-situ study with a smaller set of participants and subsequently interviewed them [20].

Method
In this more controlled study, we invited participants to use *NHistory* for one week. During this study, we used the same version of *NHistory* as for the in-the-wild study and collected the same data. Additionally, we conducted semi-structured interviews at the end of the controlled study to gain further insights. In the interviews, we asked the participants to estimate how many notifications they receive on a daily basis, how the number of notifications affects them, and how they deal with interruptions. Further, we asked them about their opinions on *NHistory* and the provided functionality.

We individually invited the participants to our lab. All participants signed a consent form, informing them about the procedure of the study and the data collection. Further, they were informed that they are allowed to withdraw their study participation at any time. We then asked them to fill out a questionnaire about demographic data. Afterwards, we introduced them to *NHistory*. We installed the app on their personal Android smartphones and walked them through the different features. We then asked them to use the app as they see fit for one week. We explicitly told them that they do not have to use it at all if they find it unnecessary. After seven full days of usage, we again invited the participants to our lab and conducted interviews with a duration of approximately 30 minutes each. Participants received EUR 15 for their participation.

Participants
We invited 17 participants to use *NHistory* for one week. One participant withdrew the study participation; thus, we excluded this participant from our evaluation. The remaining 16 participants (4 female, 12 male) were 20-36 years old (M=26.50; SD=4.07). Four of them were PhD students, eight students, two software engineers, and two teachers. All but the teachers had a technical background.

Results
The device language of the participants' smartphones was set to German in 12 cases, and to English for the other four cases. Most participants used a device running Android 6.0 (10) and 5.0 (3). Android 5.1, 7.0, and 7.1 were used once. After the day of installation, all participants used *NHistory* for seven full days to cover every day of the week.

Notifications and Apps
During the seven days of the study, the 16 participants received between 489 and 67,332 notifications. The participants had between 13 and 44 apps notifying them. In total, we logged 102,386 notifications from 147 apps. The apps that created the most notifications were *GPS Status & Toolbox* (62,597 notifications; 1 participant), *WhatsApp* (12,586 notifications; 15 participants), and the *Android OS* (6,079 notifications; 14 participants). Notification creation peaked between 6pm and 8pm. At 2am the least number of notifications were created. Participants received twice the amount of notifications on the weekend compared to the rest of the week.

Notification History Blacklist
Eight of the 16 participants made use of the option to exclude specific apps from the history. These eight participants excluded between 1 and 4 apps, totaling in 13 different apps. The apps excluded by most participants were *Google Maps* (3 participants) and the *Android OS* (2 participants).

Starred Notifications
Participants rarely used the option to star notifications. Only three of 16 participants used the feature at least once, resulting in five star events. This again shows that participants were not interested in permanently bookmarking notifications.

Snoozed Notifications
We recorded 116 snooze events from 15 participants and 33 apps. One participant did not snooze notifications. We assigned the *duration* and *point-in-time (pit)* methods evenly to the participants. Seven participants stayed with their assigned method (2 *duration*, 5 *pit*), 6 switched to the other method (4 *duration* to *pit*, 2 *pit* to *duration*), and 3 switched but returned to their assigned method (2 *duration*, 1 *pit*).

In total, 109 individual notifications were snoozed. 105 were snoozed once and 4 notifications were snoozed multiple times. We will now report the 109 initial snooze events. The 109 initial snooze events by 15 participants break down to 2 to 25 snooze events per participant (Md=5). The apps that were snoozed the most were *WhatsApp* (39 events; 11 participants), *Blue Mail* (10 events; 1 participant), *Facebook Messenger* (5 events; 4 participants), and *Facebook* (5 events; 2 participants). We again categorized all apps and conducted a Friedman test to compare the normalized data of snoozed notifications from different application categories. The results show that the application category has an influence on the users' snoozing behavior ($\chi^2(11) = 50.06, p < .001$). We conducted a post-hoc analysis with Holm-Bonferroni corrected Wilcoxon signed-rank tests. Our post-hoc analysis found no significant differences between the categories. As shown on the right side of Table 1, most snoozed notifications were from the categories *SMS/IM* (49.15%) and *Email* (13.66%).

We again saw a similar usage of the *duration* and *point-in-time* methods, and the provided options. However, in contrast to the in-the-wild study, the snooze events were more evenly distributed over the day. Peaks for snooze events can be seen at 10am, 1pm, and 5pm. Most notifications were re-triggered at 12pm, 3pm, 6pm, and 9pm. Compared to the in-the-wild study, 9am was an unpopular option to re-trigger notifications. We again found that most notifications were snoozed and re-triggered on the same day (79.82%), some on the following day (15.60%), and only few on more than two days (4.59%).

Interviews

After the participants used *NHistory* for a week, we invited them back to our lab and conducted semi-structured interviews. We used open coding for the analysis of the interviews. Three researchers coded the answers individually. Disagreements were discussed until an agreement was reached.

Handling Notifications and Interruptions

Participants estimated that they receive 15-150 notifications from 2-10 apps per day. One participant stated that he perceives the number of notifications he has to deal with as "low", eight participants perceived them as "between okay and high" and seven participants as "too high". Six participants felt "never or rarely" being interrupted by notifications, seven "sometimes", and three "often". Participants stated that they usually react immediately when they notice a notification. They stated that they attend communication-related notifications as soon as possible and other types of notifications if they have time. Silencing the phone was mentioned as a method to cope with the number of notifications participants receive on a daily basis. One participant stated that he sometimes places the phone out of reach in addition to silencing it. To cope with annoying notifications, participants mentioned simply ignoring them, dismissing them immediately, and revoking the permission to show notifications. One participant reported uninstalling apps because of annoying notifications.

Notification History and Blacklist

Five participants stated that they did not use the notification history feature because they saw no need for it. Six other participants stated that they used the history to read notifications. P4 explained that the history enables him to read longer messages than the notification drawer. He further remarked that another benefit of reading notifications in the history is that the corresponding messages are not marked as "read". Thus, chat partners do not expect him to reply immediately. Participants used the history to snooze notifications and to reflect on notifications they received during the day. P10 used the history to remember important notifications.

Nine participants stated that they saw no need to blacklist apps in the notification history. Three other participants said that they blacklisted apps which create many unimportant notifications. Examples of this kind of notifications were notifications from music apps (P16), GPS tracking apps (P4), system notifications about available Wi-Fi networks (P12), and timers (P16). These examples match the findings of the quantitative results. This indicates that these kinds of notifications are only of relevance for a limited time and participants see no point in revisiting them. Additionally, P3 excluded an app because the app generates notifications with sensitive data.

Starring Notifications

Thirteen participants found starring notifications unnecessary. P10 explained that notifications are temporary and should not be saved persistently; instead, he snoozes them if necessary. In contrast, P4 stated that he starred a notification which he described as "cool and memorable". P3 stated that he starred notifications to remember important information and tasks because it allows quick access to the notification.

Snoozing Notifications

Participants explained that they snoozed notifications at work, university, or while studying. Further contexts for snoozing notifications were being on-the-go, during sport, gaming, driving, and because they were tired.

The participants provided examples when snoozing notifications was beneficial. For instance, P2 explained that she likes snoozing notifications because she does not have to deal with the notification itself anymore if the system can remind her. Furthermore, participants found that snoozing notifications supports their attention management. P16 explained that he is less distracted if he snoozes interesting notifications to a more appropriate time. Additionally, P4 explained that snoozing notifications helps to keep the status bar clean.

Participants also told us what they disliked. Snoozing notifications manually was sometimes regarded as unnecessary, as it takes the same amount of time as, for example, answering a short message. The "snooze notification" that appeared after dismissing a notification was described as annoying. P3 explained that most of the time she does not want to snooze a notification when dismissing it. Another participant complained about the short period the "snooze notification" was shown (5*sec*) because he was sometimes too slow to snooze the notification directly in the notification center and had to open *NHistory* to snooze the notification from the history instead (P4). P2 reported that she was sometimes not able to find a notification in the history to snooze it. Participants disliked that they have to unlock their phones before they can snooze notifications from the lock screen. P11 raised concerns regarding snoozing notifications because he feels a social pressure to answer messages quickly and when a message is snoozed the sender has to wait for a reply. Another participant was concerned about being overwhelmed when snoozing too many notifications during the day to the same time slot. She suggested creating an overview of all received notifications in the evening instead.

Reasons for Snoozing Notifications

Participants mainly snoozed notifications to create reminders. For example, participants mentioned that they snoozed *SMS/IM* and *Email* notifications because they were at work and did not want to deal with personal notifications. P12 explained that she concentrated on work and wanted to receive a reminder afterwards. This was a common theme in the interviews. Participants snoozed *SMS/IM* notifications because they wanted to avoid switching their current context. Snoozing *SMS/IM* and *Email* notifications was often mentioned in regard of not forgetting a task. Another reason for snoozing notifications was that participants were sometimes not in the mood to deal with the notifications when they received them. For instance, P11 explained that he received “20-30” instant messages from a group chat and wanted to read them later. Participants also had to do other things first before being able to react to notifications. P13 explained that he received a message about a meeting with friends and had to ask his wife before he could accept the meeting. Further, participants snoozed notifications on-the-go to deal with them at home. For instance, P11 mentioned that he received a *Social* notification, which he wanted to read at home on his desktop computer. P13 explained that he snoozed an app update notification because he wanted to install the updates at home using a Wi-Fi connection. P12 snoozed a *Game* notification not to miss an in-game reward.

Duration vs. Point-in-Time

Six participants liked the *duration* method to snooze notifications, as it enables them to estimate when they will be able to deal with the notifications. P11 stated that she likes the method because it allows her to decide how long to snooze based on the current situation. Participants found the *duration* method better for short periods of time. P3 especially liked it for snoozing notifications to the same day. However, other participants stated that the *duration* method requires more cognitive effort. P2 explained that she thinks in times of the day and therefore she would have to calculate the time distance herself. Further, P7 mentioned he does not know if he has time to deal with the notification in a particular distance in time.

Eight participants stated that the *point-in-time* method supports their daily routines. For example, P7 explained that he usually knows when he can deal with notifications, e.g., in the lunch break or after work. Additionally, another participant explained that he does not have to think about how long it will take until he can deal with the notification, e.g., in the evening. Participants mentioned that they prefer the *point-in-time* method for longer distances in time, e.g., more than an hour (P10) or a week (P3). P9 stated that the pre-defined time slots are not always useful. Further, two participants found that entering a custom time is difficult when being busy. P11 mentioned that the *point-in-time* method requires a higher cognitive effort because he has to calculate the time when he wants to receive the notification.

Usefulness of *NHistory*

Eleven participants stated that the app helps them to deal with notifications, four found it somewhat useful, and one participant found it to be not useful at all. Participants liked that the app enables them to have a “clean” notification drawer, and they found the notification history helpful. P4 especially liked that he can dismiss notifications without losing them. P3 liked that she can reflect on received notifications and was interested in how many notifications she receives.

Concerns and Suggestions

Participants disliked the “snooze notification” that was shown when dismissing a notification. They suggested that, instead, the functionality should be added to the notification itself using a gesture or long-press action. Two participants would like to have the *duration* and *point-in-time* options available at the same time, and two other participants would like to customize the options. Participants disliked how multiple instant messages are grouped in notifications and snoozing a single conversation is sometimes not possible. P1 and P5 suggested that after snoozing a notification, all following notifications from the same app should be snoozed as well. Further, participants would like to snooze all notifications from all apps for a specific duration or a specific point-in-time. P15 suggested automatic rules to snooze notifications. Participants also suggested using location-based triggers instead of time-based ones. P10 wished that snoozing a notification would create a corresponding calendar event which then could be synchronized with other devices. Regarding the notification history, P4 suggested that “spammy” notifications should be automatically detected and excluded. P8 would like to see statistics about received notifications and P14 suggested an end-of-day summary of all received notifications.

DISCUSSION AND LIMITATIONS

We observed comparable usage patterns in both studies. Temporarily snoozing notifications was favored compared to permanently bookmarking them. Most snoozed notifications were of the *SMS/IM* category, followed by *Calendar/Reminder* (in-the-wild) and *Email* (controlled).

In both studies, we observed peaks in re-triggered notifications at 12pm, 3pm, 6pm, and 9pm. Additionally, in the in-the-wild study 9am was preferred. These times are likely influenced by the pre-defined options of *NHistory* but might also correlate with before and after work hours, and lunch breaks. In the controlled study, almost a quarter of the snoozed notifications were re-triggered at 6pm, likely correlating with after work hours. This was further strengthened in the interviews with many participants stating that they snoozed personal notifications at work or while studying. We observed that during the day notifications were typically deferred for short amounts of time. However, deferring notifications from the morning or noon to the evening, or from the evening until the next morning were also common use-cases. Another similarity of both studies was that the deferral of notifications was mostly limited to the same day. Few notifications were snoozed until the next day and more than two days was an exception. Still, in both studies, we observed only a small fraction of received notifications being snoozed. We assume these notifications to be of high relevance to the users but, at the same time, out-of-context and with a low urgency.

A limitation of the in-the-wild study is that we did not collect demographic data from the users. While the Google Play Store statistics indicate a diverse set of users, we have little background information. We assume that the active users were, to a certain degree, tech-savvy, as they found, installed, and configured *NHistory* without detailed instructions. In the controlled study, most participants had a technical background and were, therefore, tech-savvy as well. A second limitation is that we limited the data collection due to privacy concerns. This resulted in little knowledge about the notifications apart from which app issued them at what time. Future studies should consider context data and the notifications’ content.

While we conducted the studies, the latest version of Android was announced (March 21, 2017) and eventually released (August 21, 2017). In Android 8.0 ("Oreo") snoozing notifications was implemented natively. Swiping a notification from left to right unveils a "clock" icon. Tapping on the icon allows the user to snooze the notification for 15 minutes, 30 minutes, 1 hour, or 2 hours. This interaction is similar to the suggestions we received in the interviews. Comparing the provided options with *NHistory* and our findings, we notice the lack of long-term options for use-cases such as snoozing notifications until after work or the next morning. Since we concluded the controlled study on March 22, 2017, it is unlikely that the announcement of Android 8.0 influenced the participants. However, we assume that some users of the in-the-wild study read about the feature and subsequently searched and downloaded *NHistory* from the Google Play Store.

DESIGN IMPLICATIONS

From the findings of the in-the-wild and controlled studies, we derived design implications for future notification systems.

Consider Context and Daily Routines

Participants in the interviews reported multiple reasons for deferring notifications. For instance, when the user is focused on another task (work, studying), the notification is out-of-context (personal notifications at work), when the user is unable to attend the notification (on the go, driving, sports), or simply when the user is not in the mood to take action on the notification. Users already apply various strategies in these cases, such as muting the phone, putting it away and revoking notification permissions. Future notification systems should follow and build on these strategies.

We observed peaks in deferred notifications to, what we assume to be, before and after work hours, and lunch breaks. Especially in the controlled study, many of the deferred notifications were re-triggered in the evening. Future notification systems should consider the user's context and daily routines. For example, personal notifications at work could be automatically be detected and subsequently deferred to after-work hours. This could be combined with the creation of automatic summaries. Although we offered multiple pre-defined options to snooze notifications, users made effective use of custom durations and points-in-time. Further, participants of the controlled study suggested personalization of these options, indicating that a one-fits-all approach might not be practical.

Balance Importance and Social Expectations

Notifications from the *SMS/IM* category were snoozed most often in both studies, highlighting their importance to users. Further important categories include *Calendar/Reminder*, *Email*, and *Social*; reiterating that "notifications are for messaging" and "important notifications are about people and events" [34]. Participants stated in the interviews that they often attend communication-related notifications immediately. Especially for instant messages, there are social expectations to reply quickly [4]. Future notification systems should carefully assess the importance of a notification to decide whether or not it should be deferred and for how long.

Mind the Ephemeral Nature of Notifications

Current mobile operating systems do not display how many notifications users receive on a daily basis. We received positive feedback regarding the notification history, as it enables users to reflect on notifications. The history further allows users to "safely" dismiss notifications because they can always look them up afterward, reducing the number of pending notifications in the notification drawer. Still, participants regarded notifications as temporary. Notifications were mostly deferred to the same day or next morning. Even for notifications that were snoozed multiple times, we observed that deferring a notification for more than two days was an exception. Developers of future notifications systems that defer notifications can use this as an upper bound. Within this time span, we observed a high variance regarding the deferral duration, depending on the time of the day and the users' daily routines.

CONCLUSION

In this paper, we investigated the user-defined deferral of mobile notifications. To reduce negative effects caused by interruptions from notifications, a body of related work investigated using opportune moments for notification delivery, often based on breakpoints in the user's activity. Our approach instead allows users to manually "snooze" notifications. We focused on deriving insights into how users postpone notifications rather than eliminating negative effects directly, as snoozing a notification might even introduce a second interruption. We developed the Android app *NHistory* that extends the Android operating system in two ways. A notification history enables users to go back to previously dismissed notifications. Further, the app allows users to snooze notifications for a user-defined duration or to a user-defined point-in-time.

We explored how and why users make use of the provided functionality in a year-long in-the-wild study with 295 active users and a week-long controlled study with 16 participants. Notifications of the categories *SMS/IM*, *Calendar/Reminder*, *Social*, and *Email*, were snoozed most often. Even for notifications that were snoozed multiple times, we observed that deferring a notification for more than two days was an exception. We conducted interviews to gain insights into when and why people find deferring notifications useful. Participants mentioned avoiding context switches, especially from attending personal notifications during work. As a result, we observed a number of notifications being snoozed to before and after work hours, as well as lunch breaks. Participants raised concerns regarding deferring communication-related notifications, due to social expectations to respond as soon as possible. We conclude the paper with design implications for future smart notification systems. These systems should consider different categories of notifications, such as personal and work-related notifications, the current context of the users, as well as their daily routines. In future work, we plan to investigate snoozing notifications with user-defined rules [2], the generation of automatic notification summaries, and location-based triggers.

Acknowledgments: This work is supported by the German Ministry of Education and Research (BMBF) within the DAAN project (13N13481) and by the DFG within the SimTech Cluster of Excellence (EXC 310/2).

REFERENCES

1. Piotr D Adamczyk and Brian P Bailey. 2004. If not now, when?: the effects of interruption at different moments within task execution. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 271–278.
2. Jonas Auda, Dominik Weber, Alexandra Voit, and Stefan Schneegass. 2018. Understanding User Preferences Towards Rule-based Notification Deferral. In *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems (CHI EA '18)*. ACM, New York, NY, USA, Article LBW584, 6 pages. DOI: http://dx.doi.org/10.1145/3170427.3188688
3. Brian P Bailey, Joseph A Konstan, and John V Carlis. 2000. Measuring the effects of interruptions on task performance in the user interface. In *Systems, Man, and Cybernetics, 2000 IEEE International Conference on*, Vol. 2. IEEE, 757–762.
4. Karen Church and Rodrigo de Oliveira. 2013. What's up with whatsapp?: comparing mobile instant messaging behaviors with traditional SMS. *15th international conference on Human-computer interaction with mobile devices and services (MobileHCI'13)* (2013), 352–361. DOI:http://dx.doi.org/10.1145/2493190.2493225
5. Herbert H Clark. 1996. *Using language*. Cambridge university press.
6. Fulvio Corno, Luigi De Russis, and Teodoro Montanaro. 2016. A context and user aware smart notification system. *IEEE World Forum on Internet of Things, WF-IoT 2015 - Proceedings* (2016), 645–651. DOI: http://dx.doi.org/10.1109/WF-IoT.2015.7389130
7. Mary Czerwinski, Edward Cutrell, and Eric Horvitz. 2000. Instant Messaging and Interruption: Influence of Task Type on Performance. *Proceedings of OZCHI 2000* (2000), 356–361. DOI: http://dx.doi.org/10.1016/S1361-3723(02)01112-0
8. Android Developers. 2018. NotificationListenerService. https://developer.android.com/reference/android/service/notification/NotificationListenerService.html. (2018). [Online; accessed 20-January-2018].
9. Tilman Dingler, Dominik Weber, Martin Pielot, Jennifer Cooper, Chung-Cheng Chang, and Niels Henze. 2017. Language Learning On-the-go: Opportune Moments and Design of Mobile Microlearning Sessions. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '17)*. ACM, New York, NY, USA, Article 28, 12 pages. DOI: http://dx.doi.org/10.1145/3098279.3098565
10. Joel E. Fischer, Chris Greenhalgh, and Steve Benford. 2011. Investigating episodes of mobile phone activity as indicators of opportune moments to deliver notifications. *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services - MobileHCI '11* (2011), 181. DOI: http://dx.doi.org/10.1145/2037373.2037402
11. Sandy J J Gould, Duncan P Brumby, and Anna L Cox. 2013. What does it mean for an interruption to be relevant? An investigation of relevance as a memory effect. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 57, 1 (2013), 149–153. DOI: http://dx.doi.org/10.1177/1541931213571034
12. Joyce Ho and Stephen S Intille. 2005. Using context-aware computing to reduce the perceived burden of interruptions from mobile devices. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '05* (2005), 909. DOI: http://dx.doi.org/10.1145/1054972.1055100
13. Eric Horvitz, Paul Koch, and Johnson Apacible. 2004. BusyBody: Creating and Fielding Personalized Models of the Cost of Interruption. In *Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work (CSCW '04)*. ACM, New York, NY, USA, 507–510. DOI:http://dx.doi.org/10.1145/1031607.1031690
14. Shamsi T. Iqbal and Brian P. Bailey. 2010. Oasis. *ACM Transactions on Computer-Human Interaction* 17, 4 (2010), 1–28. DOI: http://dx.doi.org/10.1145/1879831.1879833
15. N. Kern and B. Schiele. 2003. Context-aware notification for wearable computing. *Seventh IEEE International Symposium on Wearable Computers, 2003. Proceedings.* (2003), 223–230. DOI: http://dx.doi.org/10.1109/ISWC.2003.1241415
16. Afra Mashhadi, Akhil Mathur, and Fahim Kawsar. 2014. The Myth of Subtle Notifications. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication (UbiComp '14 Adjunct)*. ACM, New York, NY, USA, 111–114. DOI: http://dx.doi.org/10.1145/2638728.2638759
17. Abhinav Mehrotra, Robert Hendley, and Mirco Musolesi. 2016. PrefMiner. *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '16* October (2016), 1223–1234. DOI: http://dx.doi.org/10.1145/2971648.2971747
18. Abhinav Mehrotra, Mirco Musolesi, Robert Hendley, and Veljko Pejovic. 2015. Designing content-driven intelligent notification mechanisms for mobile applications. *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15)* (2015), 813–824. DOI: http://dx.doi.org/10.1145/2750858.2807544
19. 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 (CHI '16)*. ACM, New York, NY, USA, 1021–1032. DOI: http://dx.doi.org/10.1145/2858036.2858566

20. Alistair Morrison, Donald McMillan, Stuart Reeves, Scott Sherwood, and Matthew Chalmers. 2012. A Hybrid Mass Participation Approach to Mobile Software Trials. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 1311–1320. DOI: http://dx.doi.org/10.1145/2207676.2208588

21. Darren Newtson and Gretchen Engquist. 1976. The perceptual organization of ongoing behavior. *Journal of Experimental Social Psychology* 12, 5 (1976), 436–450. DOI: http://dx.doi.org/10.1016/0022-1031(76)90076-7

22. Tadashi Okoshi, Julian Ramos, Hiroki Nozaki, Jin Nakazawa, Anind K Dey, and Hideyuki Tokuda. 2015a. Attelia : Reducing User ' s Cognitive Load due to Interruptive Notifications on Smart Phones. (2015), 96–104.

23. Tadashi Okoshi, Julian Ramos, Hiroki Nozaki, Jin Nakazawa, Anind K. Dey, and Hideyuki Tokuda. 2015b. Reducing Users' Perceived Mental Effort due to Interruptive Notifications in Multi-Device Mobile Environments. *Proceedings of the Joint International Conference on Pervasive and Ubiquitous Computing and the International Symposium on Wearable Computers (Ubicomp/ISWC'15)* (2015), 475–486. DOI: http://dx.doi.org/10.1145/2750858.2807517

24. Tadashi Okoshi, Hideyuki Tokuda, and Jin Nakazawa. 2014. Attelia: Sensing User's Attention Status on Smart Phones. *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing Adjunct Publication - UbiComp '14 Adjunct* (2014), 139–142. DOI: http://dx.doi.org/10.1145/2638728.2638802

25. Tadashi Okoshi, Kota Tsubouchi, Masaya Taji, Takanori Ichikawa, and Hideyuki Tokuda. 2017. Attention and Engagement-Awareness in the Wild : A Large-Scale Study with Adaptive Notifications. *Proceedings of IEEE International Conference on Pervasive Computing and Communications 2017 (PerCom '17)* (2017).

26. 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 (CSCW '17)*. ACM, New York, NY, USA, 541–554. DOI: http://dx.doi.org/10.1145/2998181.2998189

27. Veljko Pejovic and United Kingdom. 2014. InterruptMe: Designing Intelligent Prompting Mechanisms for Pervasive Applications. (2014), 897–908.

28. Veljko Pejovic, Mirco Musolesi, and Abhinav Mehrotra. 2015. Investigating The Role of Task Engagement in Mobile Interruptibility. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI '15)*. ACM, New York, NY, USA, 1100–1105. DOI: http://dx.doi.org/10.1145/2786567.2794336

29. Martin Pielot, Bruno Cardoso, Kleomenis Katevas, Joan Serrà, Aleksandar Matic, and Nuria Oliver. 2017. Beyond Interruptibility: Predicting Opportune Moments to Engage Mobile Phone Users. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 3, Article 91 (Sept. 2017), 25 pages. DOI: http://dx.doi.org/10.1145/3130956

30. Martin Pielot, Karen Church, and Rodrigo de Oliveira. 2014. An In-Situ Study of Mobile Phone Notifications. *Proc. MobileHCI '14* (2014), 233–242. DOI: http://dx.doi.org/10.1145/2628363.2628364

31. Martin Pielot, Tilman Dingler, Jose San Pedro, and Nuria Oliver. 2015. When Attention is Not Scarce - Detecting Boredom from Mobile Phone Usage. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '15)*. ACM, New York, NY, USA, 825–836. DOI: http://dx.doi.org/10.1145/2750858.2804252

32. Martin Pielot and Luz Rello. 2015. The Do Not Disturb Challenge: A Day Without Notifications. *Extended Abstracts of the ACM CHI'15 Conference on Human Factors in Computing Systems* 2 (2015), 1761–1766. DOI: http://dx.doi.org/10.1145/2702613.2732704

33. Martin Pielot and Luz Rello. 2017. Productive, Anxious, Lonely: 24 Hours Without Push Notifications. In *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '17)*. ACM, New York, NY, USA, Article 11, 11 pages. DOI: http://dx.doi.org/10.1145/3098279.3098526

34. Alireza Sahami Shirazi, Niels Henze, Tilman Dingler, Martin Pielot, Dominik Weber, and Albrecht Schmidt. 2014. Large-scale assessment of mobile notifications. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14* (2014), 3055–3064. DOI: http://dx.doi.org/10.1145/2556288.2557189

35. Angela Sasse, Chris Johnson, and others. 1999. Coordinating the interruption of people in human-computer interaction. In *Human-computer interaction, INTERACT*, Vol. 99. 295.

36. Daniel Siewiorek, Asim Smailagic, Junichi Furukawa, Neema Moraveji, Kathryn Reiger, and Jeremy Shaffer. 2003. SenSay: A context-aware mobile phone. *Seventh IEEE International Symposium on Wearable Computers, 2003. Proceedings.* (2003), 248–249. DOI: http://dx.doi.org/10.1109/ISWC.2003.1241422

37. Katarzyna Stawarz, Anna L. Cox, and Ann Blandford. 2014. Don't Forget Your Pill!: Designing Effective Medication Reminder Apps That Support Users' Daily Routines. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 2269–2278. DOI: http://dx.doi.org/10.1145/2556288.2557079

38. Dominik Weber, Alireza Sahami Shirazi, and Niels Henze. 2015. Towards Smart Notifications Using Research in the Large. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI '15)*. ACM, New York, NY, USA, 1117–1122. DOI: http://dx.doi.org/10.1145/2786567.2794334

39. Dominik Weber, Alexandra Voit, Philipp Kratzer, and Niels Henze. 2016a. In-Situ Investigation of Notifications in Multi-device Environments. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '16)*. ACM, New York, NY, USA, 1259–1264. DOI: http://dx.doi.org/10.1145/2971648.2971732

40. Dominik Weber, Alexandra Voit, Huy Viet Le, and Niels Henze. 2016b. Notification Dashboard: Enabling Reflection on Mobile Notifications. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI '16)*. ACM, New York, NY, USA, 936–941. DOI:http://dx.doi.org/10.1145/2957265.2962660

41. Fengpeng Yuan, Xianyi Gao, and Janne Lindqvist. 2017. How Busy Are You?: Predicting the Interruptibility Intensity of Mobile Users. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (CHI '17)*. ACM, New York, NY, USA, 5346–5360. DOI: http://dx.doi.org/10.1145/3025453.3025946