Design and Implementation of Notification Information Survey System and Survey Results Toward Use-side Adaptive Notification Management

Kenta Taki*, Yuki Matsuda*†, Yutaka Arakawa*‡, Keiichi Yasumoto*
* *Graduate School of Information Science, Nara Institute of Science and Technology*, Nara, Japan
Email: taki.kenta.th6@is.naist.jp, yukimat.jp@gmail.com, ara@is.naist.jp, yasumoto@is.naist.jp
† Research Fellow of Japan Society for the Promotion of Science
‡ JST Presto, Japan

***Abstract*—Many interrupt notification methods have been studied, but most of the existing research assumes that, except for the target application, the applications do not control the notification timing. However, if other applications are controlled by the same notification timing, interrupt timing will become concentrated and the effects of notification timing control may not be exerted. In addition, since the installed applications are different for each user, it is necessary to control notification timings while taking into consideration the behaviors of all the applications installed on a user's smartphone. In this research, we define notification timing control while considering the behaviors of all installed applications as "Adaptive Notification Management" and then conduct diversity surveys of notifications received by users. In this paper, we report on a system that acquires all notification information while excluding privacy data. We also report on experimental results using actual data collected using crowdsourcing and discuss how to realize the application realizing adaptive notification management.**

***Index Terms*—Notification management, Mobile application, Adaptive computing, Context awareness, Interrupt notification, Mobile survey system.**

I. Introduction

Mobile notifications provide an important means for applications to actively provide information to users, but the number of notifications users receive is increasing year by year. However, notifications provided at inappropriate timing will cause an increase in the user's stress and hamper productivity because there is a limit to the amount of information that can be perceived by humans. Therefore, many studies have been conducted on providing interruptions at optimal timing using a wide range of variables such as context [1], environment [2], and message contents [3]. By controlling the timing based on these methods, it becomes possible to improve the response rate to notifications. However, when all notifications are controlled by not only a single application but also a plurality of applications in the smartphone, one-point concentration will occur against human attention.

In order to solve this problem, it is necessary to assert control by taking into consideration the behaviors for all notifications of each user. However, the types of applications installed on a smartphone and the status of permission for reporting application notifications are different for each user. For this reason, it is difficult for application developers, who have no way to know how many different applications exist in a user's smartphone, to take them into consideration for control purposes. Even in the experimental system outlined in Okoshis research [1], which was carried out in collaboration with Yahoo, it is supposed that there will be uncontrolled normal applications in addition to Yahoo's controlled applications. Therefore, if all the other applications report at the same breakpoint, the control performance may be adversely affected. In this research, we investigate the user's behavior in relation to notifications of all applications existing on the user's smartphone, and then consider a method that can neutrally personalize the notification timing at the user side. This is called*adaptive notification management*.

In this research, in order to investigate the diversity of notifications received by users and the behaviors to each as part of efforts to achieve adaptive notification management, we developed a system that can acquire all kinds of notification information while excluding privacy data. We also used this method to investigate general users who were recruited by crowdsourcing for four weeks and collected 95,910 notification data from 20 participants.

We then analyzed the collected data set statistically, organized the tasks in the adaptive notification management function, and examined the possibility of realizing individual optimization on the user side.

II. Related work

A. Notification interrupt

The information notifications from the mobile apps can be regarded as "interrupt" for the user. When the timing of these "interrupt" is inappropriate for the user, it causes an increase in users stress and a decrease in productivity [4], [5].

This work was supported by JST Presto and JSPS KAKENHI Grant Number 16H01721.

B. Personal optimization for notifications

There is a study that analyzed the context of the notification on the user side. Due to the popularization of context-aware computing and drastic improvements to the performance of mobile terminals, it has become possible to provide services while providing fine-tuning according to the behavior and environment of each user on the user side rather than providing the same service to all mobile users, as in the past. Ho et al. proposed a personalization method of notification timing based on reinforcement learning using user context as a way to optimize notification timing for each individual [6].

C. NotificationListenerService API

On Android terminals, we can use an application program interface (API) called `NotificationListenerService` [7], with which developers can receive information such as the notification application name, text messages, and timestamps when the user receives or deletes notifications on his or her smartphone. Weber et al. developed an open source framework that more than 60, 000 users use for notification research on mobile devices [8]. In addition, Sahami et al. evaluated notifications from message applications including information on users and events, and performed notification analysis focusing on the subjectivity of the users [9].

D. Responsiveness in perspective of Notification categories

Sahami Shirazi et al. gathered about 200 million notifications from over 40,000 users. Then, by categorizing them into type of application and linking with the desktop application, they evaluated user responsiveness to those notifications objectively and subjectively. They reported that users value notifications from messengers, other communication apps, and calendars. Also, many notifications have been received around noon and at night [9] [10].

E. Purpose of this study

In this research, in order to realize adaptive notification management on the user side, i.e., maximize the response rate for each notification sent to the user, we consider a method of controlling the timing adapted to the real-time context on the user side and the response situations for all applications. This paper aims to investigate the diversity of the notifications received by the user and the more detailed response behaviors to all notifications in the smartphone, and to clarify the feasibility of adaptive notification management.

III. Survey System

A. System architecture

Figure 1 shows the system architecture of our system, named *Notification Logger*. Figure 2 shows screenshots of Notification Logger. In this experiment, we first acquire the behaviors related to receiving or deleting notifications of all the applications that the user has permitted to report through `NotificationListenerService` running in the background. Then, the behavior and location information of the users at that time is acquired from the `ActivityRecognitionAPI` [1] and the `FusedLocationProviderAPI` [2], and stored in the smartphone. After that, it adopts a mechanism to export the log data saved manually by the user. The major difference from the related research mentioned in Section II-C is that it supports logging of notification actions (e.g., notification click action, notification delete action, notification delete action for the same application), which can be acquired from the latest OS of Android 8.0 or later. Therefore, it is possible to log notification actions for which the user opened the notice and confirmed the contents, while it was only able to confirm that the notification was erased in conventional systems. In this survey, we analyze mainly the opening operation of notifications by users.

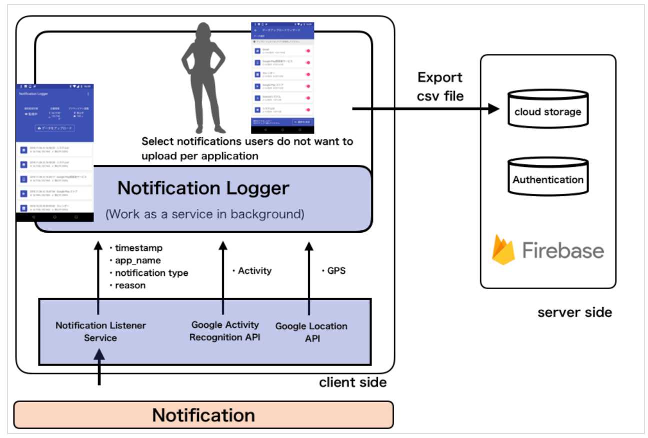

Fig. 1. system architecture

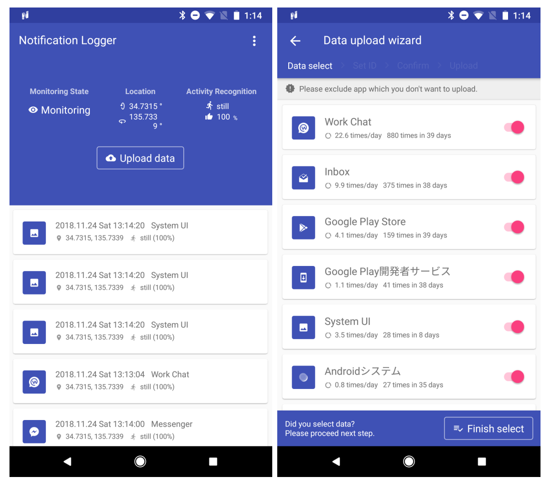

Fig. 2. Screenshots of Notification Logger

B. Attention to privacy

Table I shows data collected by the system. By using the `NotificationListenerService`, we acquire the

[1]https://developers.google.com/android/reference/com/google/android/gms/location/ActivityRecognitionApi
[2]https://developers.google.com/android/reference/com/google/android/gms/location/FusedLocationProviderApi

TABLE I
COLLECTED DATA

Data	Value
Timestamp	Time when notification action occurred
Location information*a	GPS Information (Latitude, Longitude) when notification action occurred
App name*b	Application name of notification action
Notification type*b	Posted Removed
Notification action*b (Reason code)	REASON_CLICK (1)*c REASON_CANCEL (2, 8)*d REASON_APP_CANCEL_ALL (9)*e REASON_GROUP _SUMMARY_CANCELED (12)*f
Activity recognition*g	IN_VEHICLE ON_BICYCLE ON_FOOT STILL TILTING UNKNOWN

*a Use `FusedLocationProviderAPI`
*b Use `NotificationListenerService` *c Notification click action
*d Single notification delete action. In this research, REASON_CANCEL (2) and REASON_APP_CANCEL (8) are considered to be the same
*e Group notification delete action *f All notification delete action
*g Use `ActivityRecognitionAPI`

timestamp recorded when receiving or deleting a notification, the application name included in the notification, the *notification type* (receive or delete the notification), and the *notification action* (the operation performed by the user). In addition, user location information at that time is acquired from the `FusedLocationProviderAPI` and user actions are acquired from the `ActivityRecognitionAPI`.

Although the text messages included in the notifications could also be acquired, we intentionally excluded the contents of those message from this experiment in order to lower the hurdles imposed on the participating users. Nevertheless, they still include personal information data such as the application name, position information, and action information. That is why we developed a system that does not automatically upload the log data to the server, but instead stores them in the smartphone to protect the user's privacy. The users themselves can check the log data before uploading data in this system.

IV. Survey experiment

A. Investigation method

In this experiment, we recruited test participants using "Crowdworks", which is a major Japanese crowdsourcing service. We conducted data collection experiments for 20 applicants for four weeks. Participants installed the "Notification Logger" mentioned in Section III on their normally used Android smartphones and gave permission to acquire the position information, sensor information, and the state and time spent in everyday life where notifications are monitored. We sent reminders to participants every week and uploaded data on four occasions.

TABLE II
EXPERIMENT PARTICIPANT ATTRIBUTE

			Number
Breakdown of participant attribute	Gender	Male	7
		Female	13
	Age	20–29	8
		30–39	9
		40–49	3
	Profession	Employee	9
		Housewife	6
		Part-time job	3
		Student	2
Total number			20

TABLE III
RESULT OVERVIEW

	Value	Result
Overview	Total number	95, 910
	Post count	68, 261
	Removed count	27, 649
Breakdown of notification action	Notification click	3, 506
	Single Notification delete	12, 490
	Group Notification delete	3, 643
	All Notification delete	92
	Notification delete by status bar	5, 144
	All Notification delete by status bar	2, 677
	Notification Package change	83
	Notification channel ban	6
	Notification timeout	8

B. Investigation conditions

We adopted only applicants who are using terminals equipped with Android 8.0 and above because the `NotificationListenerService` API used in Notification Logger can only get the behavior of notification tap (Click, Opening) in Android operating system (OS) version of Android 8.0 or higher. Table II shows the participants attributes. The participants were 20 male and female volunteers with ages ranging from 20 to 49.

V. Results and discussion

Table III gives an outline of the experiment results. In this experiment, 95,910 notification data were acquired over four weeks.In the breakdown of the notification data, the number of notifications received was 68,261, and the number of notifications removed was 27,649. Of those, 3,506 removal cases involved clicking (opening) the notification, and the 24,143 cases involved removal via the delete function of the notification.

A. Analysis result in perspective of app category

1) User's action for categorized app: We consider the details of user actions for each categorized app. First of all, we categorized apps by their main function as follows: *messenger*, which are communication apps such as LINE and facebook

2*b

TABLE IV
NUMBER OF USER ACTION WITHIN CATEGORIES

	post	click	single delete	group delete	all delete	status bar delete	status bar all delete	others
messenger	17, 854	1, 559	659	2, 931	0	1, 236	211	11
mail	18, 902	712	2, 748	1, 651	1	1, 412	973	14
social	6, 001	263	58	1, 266	0	636	219	9
money	3, 946	746	81	251	1	462	270	17
game	1, 132	82	4	17	78	279	77	7
news	770	4	0	4	0	15	111	4
map	3, 964	22	0	331	0	44	102	7
shop	664	23	2	17	1	147	77	5
system	13, 209	72	77	4, 132	7	486	172	4
browser	1, 552	5	0	196	0	146	364	6
payment	45	0	0	0	0	14	8	0
music	100	3	0	1	0	2	39	1
phone call	284	3	0	103	0	5	11	0
life log	21, 068	2	0	27	1	4	2	0
other	6, 419	10	14	1, 563	3	256	41	12

messenger; *mail*, such as e-mail apps; *social*, which are social networking service (SNS) apps such as Instagram and Twitter; *money*, such as crowdsourcing and flea market apps; *game*, such as social game apps; *News*, such as Yahoo news and Gunosy; *map*, such as Google Maps; *shop*, such as store coupon apps; *system*, which is the system apps of Android OS; *browser*, such as Google chrome; *payment*, such as cashless payment apps; *music*, such Google play music and Spotify; *phone call*, which is phone call; *lifelog*, such as Strava and diet apps; and *others*.

Table IV shows the number of detailed user actions for each category. *Messenger* had the most click actions, and there were many deletions for each group. *Mail* has more posts than *Messenger*, but few clicks and many single deletions. From this, it can be considered likely that there are many *Mail* notifications that users do not want to open. The number of clicks to *Money* post was large, and the response rate was the highest. This indicates that money is still interesting, and it can be considered that this category is easy to report.

Many data included notifications from the OS apps, however, there were many phenomena that were not displayed as notifications on the smartphone or where the `NotificationListenerService` caught the log repeatedly as if it received multiple notifications during the downloading operation of the apps etc. Also, since the life log has the largest number of posts, but removable actions were hardly seen, there is a possibility that a mechanism is implemented to keep applications running by posting silent notifications in the background for logging purposes.

2) Response time until click action: Next, we considered the response time from receiving the notification until the application is clicked.

Figure 3 shows a histogram of the response time from the reception of the notification until it is clicked. From the figure, we can see that most of the notifications are being clicked within about one hour (3,600s) from the time they were received. Hence, we analyzed the notification clicks that are responded to within one hour.

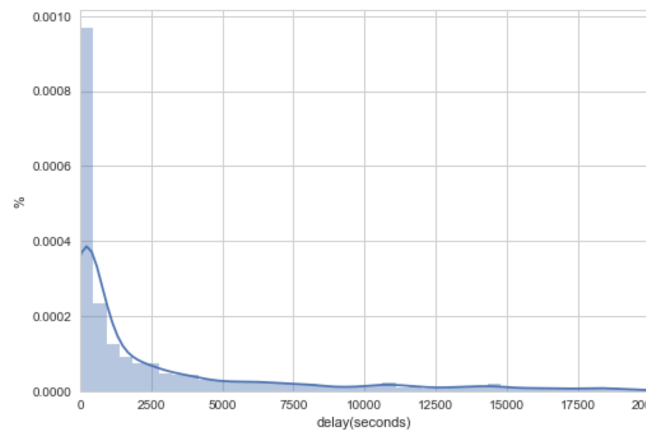

Fig. 3. Histogram of response time until the notification click action.

3) Inducing of click action by succeeding notifications: As shown in Figure 4, the click event by the user is not necessarily caused by the application notice itself. For example, we should also consider the possibility that the user can click the notification, which the user already received from "app y", at the timing of receiving a notification from "app x". Table V shows the aggregated results of the number of notifications when the last received notification is sent from the same application and different applications for one notification are clicked by the user. The results show many notifications were received from other applications just before clicking. Therefore, when predicting the click rate, it can be suggested that it is effective to consider what notifications arrived just before the click was induced.

TABLE V
RELATIONSHIP BETWEEN CLICKED NOTIFICATION AND ITS JUST BEFORE NOTIFICATION.

App type	Clicked notification and its just before notification is:	
	same app	**different app**
messenger	1075	384
mail	181	456
social	68	139
money	157	516
game	4	66
news	0	46
map	6	15
shop	0	21
system	36	28
browser	1	4
payment	0	0
music	2	1
phone call	1	1
lifelog	0	2
Others	3	5
Total	**1, 534**	**1, 684**

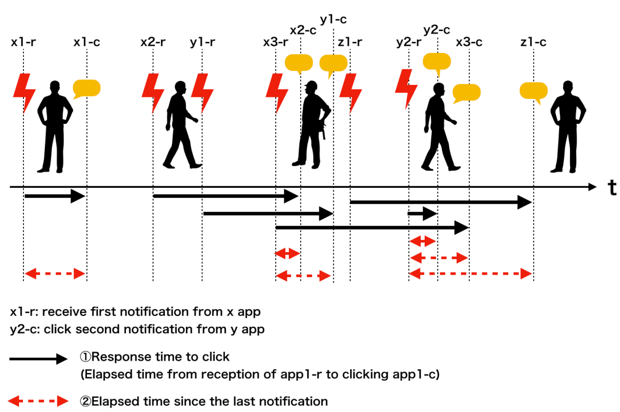

Fig. 4. Response time since the last notification.

B. Analysis result in perspective of spot

1) Click through rate for each spot: In order to analyze user's position information for each fixed range, conversion from the acquired latitude and longitude to a scale called *GeoHex* was performed. This is a scale used to fill the world surface with hex (regular hexagons) without gaps and to express all the points in the world [11]. In this study, we converted the level of GeoHex to Level 6, which makes the center distance about 2.7 km using the GeoHex python library [3]. We sorted the ID as the spots where the user stayed frequently in the descending order of the number of data with respect to the spot after conversion. The `id1` represents the spot where each user stayed most frequently, `id2` represents the spot where the user stayed second most frequently. Figure 5 shows RR at the top five spots in terms of stay frequency.

[3]https://pypi.org/project/py-geohex3/

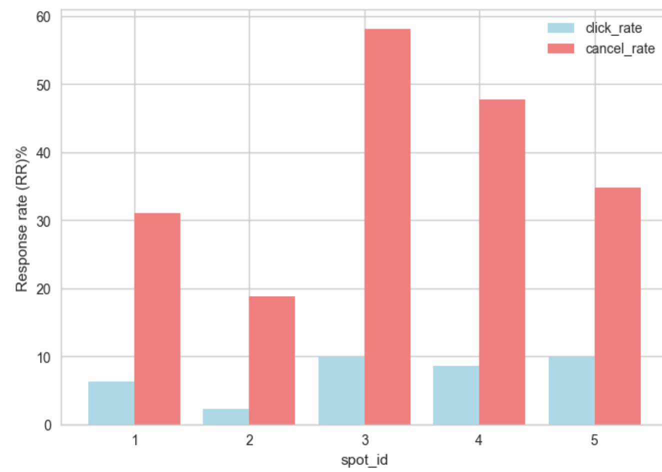

Fig. 5. Response rate at the top five spots.

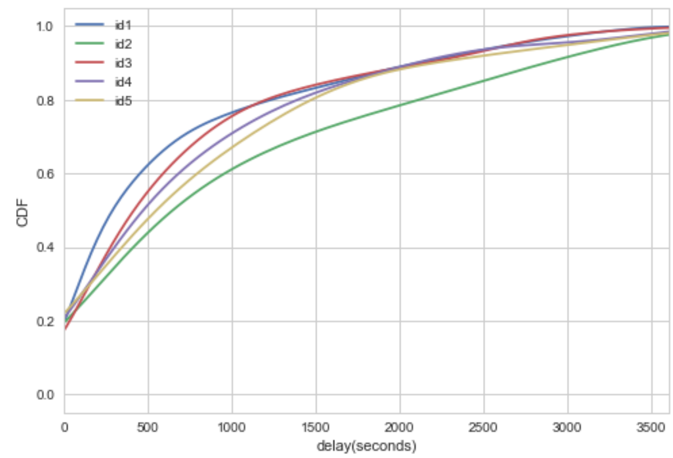

Fig. 6. CDF on users' response time to notification at the top five spots.

From the graph, it can be seen that the CTR is the lowest despite `id2` being the place among all the spots where the user stayed second most frequently.

2) Response time per spot: The CDF of response time until the notification click at each spot is shown in Figure 6. It was found that the second most frequently used spots are the most difficult locations for responding to notifications since `id2` has the longest response time to click at those spots.

3) Relationship between users and spots: Figure 7 shows the relationship between the spots with the top five stay frequencies and the time slot. From the distribution of spots by time slot, it is seen that `id1` is a spot where the user often stays at midnight and `id2` is statistically a spot where the user frequently stays during the daytime. Therefore, it can be considered likely that `id1` is the user's home and `id2` is the user's workplace or school.

C. Analysis result in perspective of time slot

Figure 8 shows the CTR per time slot. It can be seen that The number of notifications received has increased from 12:00

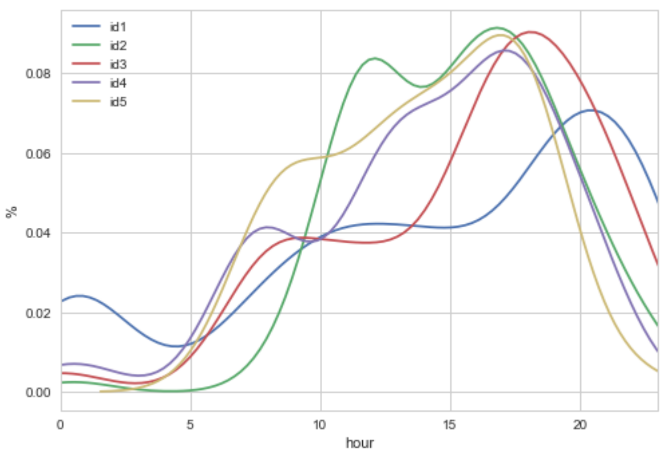

Fig. 7. Distribution of the spot for each time slot.

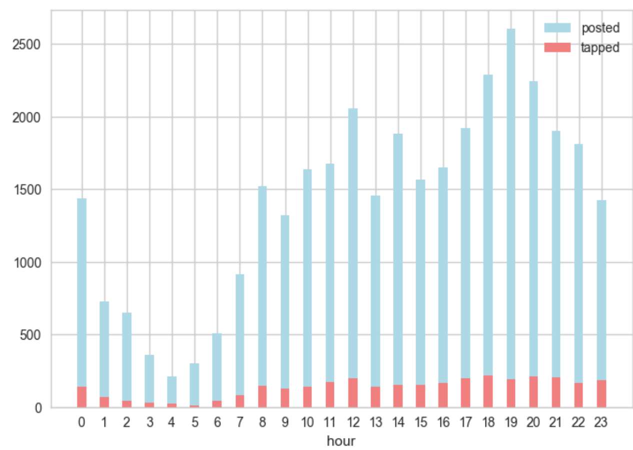

Fig. 8. Number of receiving and clicking of notifications for each time slot.

during the day and from 18:00 to 20:00. However, there is not much change in the number of clicks until bedtime. Therefore, there might be an upper limit to the number of notifications the user can click on at each time, and it can be considered likely that there is a high possibility that the CTR will be lowered even if notifications over the number of upper limit are sent.

VI. For adaptive notification management

In this study, the following results were confirmed:

- Detailed actions for categorized app seem to be effective as feature values.
- Statistically, the spot where the user stayed second most frequently has a high correlation with responsiveness.
- It was confirmed that there is an upper limit to the number of responses to notifications per unit time.

In addition, from Table V, we consider it possible to propose a control method to more effectively induce notification responses by clarifying the correlations between the clicked and previous notifications.

VII. Conclusion

In this research, in order to realize adaptive notification timing control, we developed a system that can safely investigate notifications received by users, examined the diversity of all notifications, and looked at the behavior of users in response to each notification. Specifically, we collected 95,910 notifications data from 20 participants recruited by crowdsourcing through four weeks of survey experiments, and statistically analyzed them.

By categorizing the applications, characteristic click behaviors and detailed deletions were found for each category. In addition, it was confirmed that the notification responsiveness was greatly decreased at the spot where the user stayed second most frequently, and it was confirmed that the number of responses to human notifications still had an upper limit. In this research, it was newly discovered that many notification opening actions were induced by notifications from other applications. By clarifying the correlation between notifications, we aim to realize an adaptive notification timing control system on the terminal side that is close to the user.

Acknowledgments

This work is supported by JST PRESTO.

References

[1] T. Okoshi, K. Tsubouchi, M. Taji, T. Ichikawa, and H. Tokuda, "Attention and engagement-awareness in the wild: A large-scale study with adaptive notifications," *2017 IEEE International Conference on Pervasive Computing and Communications, PerCom 2017*, pp. 100–110, 2017.

[2] D. Weber, A. Voit, P. Kratzer, and N. Henze, "In-situ investigation of notifications in multi-device environments," *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '16*, pp. 1259–1264, 2016.

[3] A. Mehrotra, M. Musolesi, R. Hendley, and V. Pejovic, "Designing content-driven intelligent notification mechanisms for mobile applications," *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing - UbiComp '15*, pp. 813–824, 2015.

[4] E. Cutrell, M. Czerwinski, and E. Horvitz, "Notification, disruption, and memory: Effects of messaging interruptions on memory and performance," *Proceedings of the INTERACT 2001*, no. 1999, pp. 263–269, 2001.

[5] G. Mark, S. T. Iqbal, M. Czerwinski, P. Johns, A. Sano, and Y. Lutchyn, "Email Duration, Batching and Self-interruption," *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16*, pp. 1717–1728, 2016.

[6] B.-j. Ho, L. Angeles, M. Koseoglu, and L. Angeles, "Nurture : Notifying Users at the Right Time Using Reinforcement Learning," 2018.

[7] "Android developers. 2018. notificationlistenerservice. https://developer.android.com/reference/android/service/ notification/notificationlistenerservice.html."

[8] D. Weber and A. Voit, "Notification Log : An Open-Source Framework for Notification Research on Mobile Devices," pp. 1271–1278.

[9] A. Sahami Shirazi, N. Henze, T. Dingler, M. Pielot, D. Weber, and A. Schmidt, "Large-scale assessment of mobile notifications," *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*, pp. 3055–3064, 2014.

[10] M. Pielot, K. Church, and R. D. Oliveira, "An In-Situ Study of Mobile Phone Notifications," pp. 233–242, 2014.

[11] @sa2da, "GeoHex - Hexagonal Geo-coding System," https://sites.google.com/site/geohexdocs/.