Investigating Interruptibility at Activity Breakpoints using Smartphone Activity Recognition API

Mikio Obuchi
Keio University
5322 Endo, Fujisawa,
Kanagawa, Japan
fabius@ht.sfc.keio.ac.jp

Jin Nakazawa
Keio University
5322 Endo, Fujisawa,
Kanagawa, Japan
jin@ht.sfc.keio.ac.jp

Wataru Sasaki
Keio University
5322 Endo, Fujisawa,
Kanagawa, Japan
wataru@ht.sfc.keio.ac.jp

Hideyuki Tokuda
Keio University
5322 Endo, Fujisawa,
Kanagawa, Japan
hxt@ht.sfc.keio.ac.jp

Tadashi Okoshi
Keio University
5322 Endo, Fujisawa,
Kanagawa, Japan
slash@ht.sfc.keio.ac.jp

Ubicomp/ISWC'16 Adjunct , September 12-16, 2016, Heidelberg, Germany.
©2016 ACM. ISBN 978-1-4503-4462-3/16/09...$15.00.
DOI: http://dx.doi.org/10.1145/2968219.2968556

Abstract
We propose a system for improving the answer rate to ESM inquiry to reduce the user's mental burden by detecting breakpoints in user's physical activity and pushing notifications in such timings. We conducted an in-the-wild user study with 30 participants for 4-days. The results revealed the effectiveness of breakpoint-based notification delivery. In the best case, 70.0% improvement in user's response time to notifications was observed at a transition to the user's activity from "walking" to "stationary".

Author Keywords
interruptibility; notification; user attention; activity recognition; smartphone

ACM Classification Keywords
H.3.4 [User profiles and alert services]: Systems and Software

Introduction
Recently, as lots of sensor data can be obtained by spreading smartphones and wearable devices, many research works on human activity recognition have been actively conducted [2, 7] on top of such data. Further recent works have also been challenging to sense human's inner state, such as emotion, feeling, and mood [8]. In such work, researchers regularly gather users' annotation labels on the

sensor data, very often through ESM (Experience Sampling Method), in order to use them combinedly as training data in machine learning-based model training. However, from the viewpoint of both researchers and subjects, users answering their own inner status for many times in one day would be a big burden. In this research, we especially focus on recent activity recognition API [1] provided by smartphone OSs (very often with dedicated "Motion co-processor" support) and investigate interruptibility of the users by using output from such API. We propose a system for improving the answer rate to ESM inquiry to reduce the user's mental burden, by means of detecting "breakpoint" [9] in user's physical activity and pushing notifications in such timings. To the best of our knowledge, our work is the first interruptibility investigation based on the physical-activity breakpoint detection implemented on top of the recent activity recognition API provided by smartphone OS. As evaluation, we conducted an in-the-wild user studies with 30 participants for 4-days. We confirmed reduction in users' answer time after receiving notification and improvement of response rate.

The contribution of our paper is three-fold.

1. Investigation of interruptibility based on the physical-activity breakpoint detection that is implemented on top of activity recognition API provided by smartphone OS (iOS).
2. The design and implementation of a system for finding breakpoints in user's physical activity on top of activity recognition API on smartphones.
3. Our in-the-wild user studies with 30 participants for 4-days, showing that user's interruptibility is affected by several types of breakpoints.

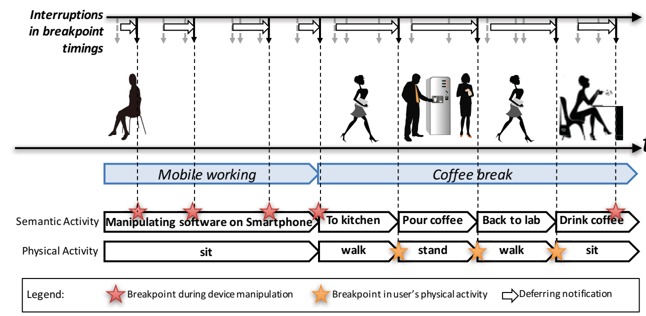

Figure 1: Breakpoints in user's daily lives in ubiquitous computing. If the user is manipulating his/her local device or not, notifications will be issued to the user. Several past literatures have been showing that, by deferring delivery of the notification until breakpoints (boundary of user's activities), the user's mental effort gets lowered.

ESM and Interruptibility

Deployed with a rich set of embedded sensors, smartphones are widely spread in our daily lives. Since such ubiquity, smartphones are actively used for many mobile sensing researches as well as for annotation of the ESM [4] questionnaires. ESM is a research method that asks participants to make notes on their experience in real-time, and is quite commonly used for ground truth labeling for mobile sensor data. According to the researcher's experiment plan, the experiment participants receives push notifications on ESM for several times a day.

However, due to several reasons including user's physical unavailability on the phone, high attentional workload (when additional burden for answering ESM is beyond his/her workload capacity), there are certain cases where the participants cannot answer the ESM. That absence in ESM leads a reducing number of sampling data and is not beneficial from the researchers' perspective, either. If we can

Figure 2: Screenshot of ESM in MyFactor. User sees a question asking the user's inner states. The user answers with one of the 5-Likert scale options.

systematically and slightly delay such ESM notifications to a timing that leads lower burden for user's answering and higher answer rate **within a certain time window allowed by the experiment design**, that would be beneficial to both researchers and the experiment subjects. For such timing, we focus on user's breakpoint in this research. Various existing literatures [6, 5, 3] revealed that notifications delayed until breakpoint (a boundary between user's activity, shown in Figure 1) resulted in lower cognitive load compared to those sent out "immediately". Previously, Attelia II [10, 11] revealed that notification delivery in breakpoint timings detected in real-time on smartphones and smart watches reduced user's mental workload. Attelia used two types of breakpoints, namely user-device interaction-based breakpoints and physical activity-based breakpoints. Meanwhile, in this research, we particularly focus on breakpoints in user's physical activity that can be detected by use of activity recognition API which the latest smartphone OS provides as one of the "common" functionalities for versatile smartphone applications. Also, previously Attelia used a physical activity breakpoint model built upon user's questionnaire results. Our system, on the other hand, gathers real sensor data of all possible changes in physical activites and investigates which type of activity change can be used as a useful breakpoint.

ESM with Breakpoint-based Notification Delivery

For this research, we have integrated physical activity breakpoint-base notification delivery mechanism into our lifelog collection system "MyFactor". MyFactor is our research-purpose lifelog collection system that consists of an iOS smartphone application and the backend server. The mobile application consistently record various types of embedded sensor data inside the phone and uploads them into our secure server. Meanwhile, occasionally the application asks the user of his/her inner status (such as mood and emotion) by using ESM method. The server issues a push notification to each client to notify the user. When user responses, a screen shown in Figure 2 appears to record user's status.

Activity Recognition API in iOS

In our physical activity-based breakpoint detection mechanism, we rely on Apple's CoreMotion activity recognition API "CMMotionActivityManager" [1] which is available in iOS7.0 or later. Using the "M" series motion co-processor on iPhone 5S or later, this API internally uses several sensors (supposed to be accelerometer, GPS, and others) and outputs user's current activity with 6 different labels, "stationary", "walking","running", "cycling", "automotive", and "unknown". The output value is a boolean vector of 6 types. In some cases, multiple activities possibly can be "true" (such as a case both "stationary" and "automotive" are true when the user's driving car stops at a signal). The API outputs the value when a change of activity is detected, not with a constant periodicity. Also, each output value includes a confidence value of the detection with three degrees (high, medium, and low).

Physical Activity-based Breakpoint Detection

In our integration, we designed to use such outputs from the CoreMotion API and generally to treat a change in the user's activity as a physical activity-based breakpoint. For example, breakpoints would be detected when the user changes from "stationary" to "walking", or changing from "automotive" to "automotive, stationary". Inside the MyFactor iOS application, this logic continuously executes and keeps detecting user's breakpoints. We confirmed that, thank to "M" co-processors, power consumption overhead is really trivial in this methodology. Note that, in our design, we omit the label "unknown". We also decided to use only "high" and "medium" confidence results due to overwhelming results with "low" confidence.

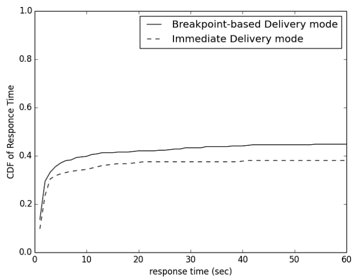

Figure 3: CDF: Response Rate and Response Time (Notification to Response) (Y-axis: maximum 1 minute

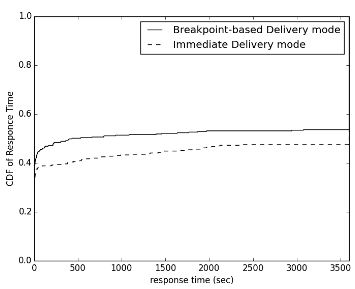

Figure 4: CDF: Response Rate and Response Time (Notification to Response) (Y-axis: maximum 1 hour

In-the-wild Evaluation

We conducted an in-the-wild user study with 30 participants for 4-days to evaluate the effectives of notification delivery in breakpoint timings.

Participants

The participants are university (undergraduate and graduate) (20 male and 10 female) students of ages 18–26. We installed our MyFactor iOS application into their own daily-using iPhones.

ESM

During the experiment, the participants were asked to annotate the ESM questionnaires when they receive a notification on their smartphones. The ESM questionarries that users were asked to annotate during the experiment were 3 types; (a)stress, (b)busyness, and (c)tiredness. The question was randomly selected each time users react to the notification. The users were asked to answer the question with 5-point Likert scale. The notification was issued only during 9:00am to 9:00pm daily.

Result and Analysis

Notification Delivery Modes

During the 4-day experiment, each participant experienced "immediately delivery"-style notification delivery mode (i.e., the situation without our breakpoint mechanism) for 2 days, and "breakpoint-detection"-based notification delivery mode for 2 days. The selection and the order of the daily mode change was randomized, and not shown to the users. In the former mode, each notification was displayed with 1 hour and random padding (maximum 5 minutes) period. In the latter mode, each notification was delayed until user's immediate breakpoint (detected in real-time) although the basic "hourly" interval is the same as the former mode.

Our objectives in this study were as follows:

	Total	Delivery in Breakpoint	Immediate Delivery
Number of Notification	778	397	381
Number of Annotation	423	231	192
Response Rate	54%	58%	50%
Average Response Time	603.3s	641.9s	556.9s

Table 1: Number of notifications, annotations answered, and average response time

1. Investigate the occurrence rate of changes of activities.
2. Investigate which type of activity change is relevant for reducing user's response time.

Successful Annotation Rate

The number of notifications and annotations obtained during the experiment is shown in Table 1. The annotation rate in "delivery in breakpoint" mode was 58%, this value is higher than the "immediate delivery" mode (50%).

Response Time and Rate

Figure 3 and Figure 4 are the CDF graphs of response time and response rate in "immediate delivery" and "breakpoint-based delivery" modes. Figure 4 shows the graph with the span of 1 hour after each notification is issued. On average, we had another notification issued 1 hour later, thus we have 1 hour as a (longest) "timeout" for each notification. The answer rate in breakpoint-based mode is always higher than that in the immediate delivery mode. In other words, users always responded faster in the breakpoint-based mode.

Detected Breakpoint

On the breakpoint detection, during the 4-day experiment, 28 participants who successfully recorded the data detected 20660 breakpoints as total. In other words, each user detected 184.5 breakpoints every day. Table 2 shows the occurrence rate of activity changes that proceeded

“From” Activity	“To” Activity	Ocuurance Rate
automotive	stationary, automotive	31.8%
stationary, automotive	automotive	30.4%
walking	stationary	6.7%
stationary	automotive	6.1%
stationary	walking	5.9%
automotive	stationary	3.8%
automotive	walking	2.2%

Table 2: Type of Activity Breakpoint and Occurrence Rate

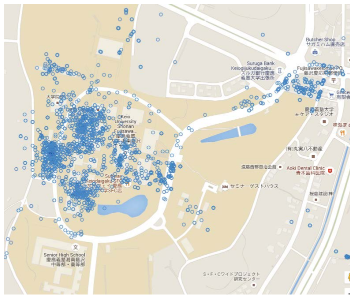

Figure 5: Breakpoint Map

the breakpoints. The most frequently detected breakpoint was from “automotive" to “stationary, automotive" (31.8%). According to Apple’s API document, this type of activity change can occur, for example, when the vehicle that the user is riding has stopped. The top 2 breakpoint types, related to “automotive", is more than 60% of total breakpoint detected. Although, we need further investigation on this, considering that all the participants are students most of who commute via train and bus (usually 1-3 hours a day for both ways), this may be due to the current implementation and/or configuration of activity recognition mechanisms that can possibly be quite sensitive to these activity modes.

Breakpoint Types and Effectiveness

Table 3 summarizes the detected breakpoint types, the average notification response time of that breakpoint type, and the comparisons of the response time against the average response time in “immediate delivery” mode. In the “Reduction” column, positive numbers means that average response time is reduced in the specific breakpoint type case. The most interesting thing we found that is, breakpoint type “from walking to stationary” resulted in 70.0% improvement in response time, while the opposite “stationary to walking” breakpoint resulted in 37.5% degradation. Again, results in breakpoint types including “automotive” are varying for each type. These types probably needs further investigation. We guess that the certain time is needed to

“From” Activity	“To” Activity	Average Response Time	Reduction of Response Time	Count
stationary	walking	765.9s	-37.5%	77
stationary	automotive	863.4s	-55.0%	28
stationary, automotive	automotive	483.0s	13.3%	25
walking	stationary	166.9s	70.0%	20
automotive	walking	309.5s	44.4%	10
cycling	walking	285.8s	48.7%	9
walking	automotive	4.6s	99.2%	9

Table 3: Type of Breakpoint and Reduction of Response Time

obtain the trustworthy activity label and it means the confidence of activity recognition and real-time breakpoint sensing are in a trade-off relationship. Being notified after a few seconds passed may not be an actual breakpoint timing since it must be the boundary of activities. Therefore, we think that possible cause of some activity changes that increased the response time is the notifications issued to the users during their walking and driving activities in such case of activity changes from “stationary" to “walking" or from “stationary" to “automotive".

Conclusion and Future Work

As a result of 4-day in-the-wild experiment with 30 participants, our system that delivers a notification in breakpoint of user’s physical activities showed improvement of a response rate and time, compared with the conventional “immediately delivering” notification approach. We still need to investigate and determine which activity change (breakpoint type) actually reduces the response time and increase the annotations.

Our future work is also in the integration of interruptibility research with other research areas and applications, such as smart city research. Figure 5 shows a breakpoint map (in our university campus) that plotted the location of breakpoints detected during our user study. We can observe that breakpoints are likely to be plotted in places such as laboratory buildings where the participants often stay, intersections, and bus stops in university camps. Sharing this type of information can enable the global "interruptibility API in smart cities".

Acknowledgement
This work is partly supported by Ministry of Education, Culture, Sports, Science and Technology, Japan. This work is also partly supported by a joint research project with Yahoo Japan Corporation.

REFERENCES

1. Apple Inc. 2014. CMMotionActivityManager. `https://developer.apple.com/library/ios/documentation/CoreMotion/Reference/CMMotionActivityManager_class/index.html`. (2014).
2. Ling Bao and Stephen S Intille. 2004. Activity recognition from user-annotated acceleration data. In *Pervasive computing*. Springer, 1–17.
3. Joel E. Fischer, Chris Greenhalgh, and Steve Benford. 2011. Investigating Episodes of Mobile Phone Activity As Indicators of Opportune Moments to Deliver Notifications. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI '11)*. 181–190. DOI: `http://dx.doi.org/10.1145/2037373.2037402`
4. Joel M Hektner, Jennifer A Schmidt, and Mihaly Csikszentmihalyi. 2007. *Experience sampling method: Measuring the quality of everyday life*. Sage.
5. Joyce Ho and Stephen S. Intille. 2005. Using Context-aware Computing to Reduce the Perceived Burden of Interruptions from Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '05)*. 909–918.
6. Shamsi T. Iqbal and Brian P. Bailey. 2007. Understanding and Developing Models for Detecting and Differentiating Breakpoints During Interactive Tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '07)*. 697–706.
7. Eunju Kim, Sumi Helal, and Diane Cook. 2010. Human activity recognition and pattern discovery. *Pervasive Computing, IEEE* 9, 1 (2010), 48–53.
8. Robert LiKamWa, Yunxin Liu, Nicholas D Lane, and Lin Zhong. 2013. Moodscope: Building a mood sensor from smartphone usage patterns. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*. ACM, 389–402.
9. Darren Newtson and Gretchen Engquist. 1976. The perceptual organization of ongoing behavior. *Journal of Experimental Social Psychology* 12, 5 (1976), 436–450.
10. 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. In *Proceedings of IEEE International Conference on Pervasive Computing and Communications 2015 (PerCom '15)*.
11. 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. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM, 475–486.