Towards Attention-aware Adaptive Notification on Smart Phones

Tadashi Okoshi, Hiroki Nozaki, Jin Nakazawa, Hideyuki Tokuda

slash@ht.sfc.keio.ac.jp
Graduate School of Media and Governance, Keio University
5322 Endo, Fujisawa, 252-0822, Japan
Tel:+81-466-47-0836, Tel:+81-466-47-0835

Julian Ramos, Anind K. Dey

Human Computer Interaction Institute, Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213

Abstract

As the amount of information to users increases with the trends of an increasing numbers of devices, applications, and web services, the new bottleneck in computing is human attention. To minimize users cognitive load, we propose a novel middleware "Attelia" that detects breakpoints of user's mobile interactions to deliver notifications adaptively. Attelia detects such timings in real-time, using only users phones, without any external sensors, and without any modifications to applications. Our extensive evaluation proved Attelias effectiveness. In-the-wild user study with 30 participants for 16 days reduced users cognitive load by 33% in users real smart phone environments.

Keywords: attention-awareness, interruptibility, notification, mobile sensing, middleware

1. Introduction

Since the introduction of the computer, users' attention has remained constant; however, the amount of information from computer systems, applications, and services has been growing. This trend has driven users to perform more multi-tasking and depend more on notifications for completing computing tasks.

on their devices. As a result, they face an increasing number of interruptions caused by the notifications. The number of connected computer devices, including a users own mobile and wearable devices, as well as sensors, cameras, servers and other devices embedded in the environment (such as home, office or city space) have been increasing as well. Users possess, carry and utilize an increasing number of mobile and wearable devices, such as smart phones, smart watches or smart bands [1], sometimes interacting with multiple of them simultaneously [2]. Driven by several technological and market trends, such as the web and cloud platform technologies that enable agile development and deployment of applications and services (through channels like the "AppStore"), each of the users devices is loaded with more applications and back-end services. Furthermore, users have been communicating more with their peers, as new types of communication channels on the web such as social networking or location-based networking, have emerged.

Focusing on this "information overload", human attention, rather than computation power or network bandwidth, is the new bottleneck in computing [3]. In this paper, we focus on the particular problem of "interruption overload", a type of distraction caused by an undue amount and inappropriate delivery of interruptive notifications from computer systems. Conventional notification systems that deliver notifications immediately after they are available, have negative impacts on users' work productivity [4, 5, 6, 7]. One solution is to defer such notifications until the transition between a user's two contiguous activities is detected. This interval of time or transition in between activities is referred to as a "breakpoint" [8] in the psychology literature. This deferral can reduce the negative effect on users' cognitive load made by the interruption.

In this work we especially focus on the "mobile experience" of users on a smart phone while they are actively manipulating their devices and show how our system can detect breakpoints during such periods, and defer notifications to these breakpoints. This work is a first step towards "user-attention-aware" adaptation in notification systems. Our system, Attelia features (1) its ability to work well (as a lightweight background process) on smart phone devices, (2) applicability to any type of application installed on the devices, (3) real-time breakpoint detection in order to realize real-time adaptation, and (4) needing only the smart phone, and not any external psycho-physiological sensors such as an ECG or EEG monitor.

A controlled in-lab user study with 37 users validated that, for users who showed greater sensitivity for interruptive notification timings, notifications displayed at the timing of breakpoints, detected by Attelia, resulted in 46% reduction in user's cognitive load, compared with the notifications displayed at the conventional "random" timings. A further "in-the-wild" user study with 30 users for 16 days, conducted after the promising results of the in-lab user study, validated the effectiveness of Attelia in a real-world situation. For those who showed greater sensitivity for interruptive notification timings, notifications displayed at the timing of breakpoints resulted in 33% lower cognitive load. Furthermore, response time of the users to the notifications (displayed in the breakpoint timings) was quicker by 13% than the notifications in the random timings.

The contributions of this paper are two-fold.

1. The design and implementation of Attelia, our novel mobile middleware on smart phones that detects user's breakpoint timing in real-time, without any external psycho-physiological sensors.
2. The results both from in-lab and in-the-wild user studies which validated the effectiveness of Attelia for reducing users' cognitive load.

The paper is structured as follows. Section 2 introduces the problem of interruption overload by notifications. Section 3 introduces several recent trends in notifications and specifies the requirements for the adaptive scheduling of notifications on smart phones. Section 4 shows our approach for designing Attelia, followed by Section 5 that describes its system architecture. We show the results from our controlled user study in Section 6, and the results from the in-the-wild user study in Section 7. Based on the analysis of the user studies, Section 8 discusses future research opportunities. Section 9 introduces related work. Finally, Section 10 concludes the paper.

2. Interruption Overload

The excessive number and ill-timed delivery of interruptive notifications cause interruption overload, one piece of the larger information overload problem. For this problem, more research has been focusing on the topics of interruption and multitasking [9].

Notifications from computer systems are the main source of interruption in a user's computing environment. The concept of notification was originally for delivering information to users in a more speedy and timely manner, rather than requiring users to manually pull new information.

Although the notification system gives such merit to users, past researches have shown that excessive numbers of notifications or ill-timed delivery of notifications results in several types of negative impacts on user's work. These impacts include reduction in user's work productivity [4, 5, 6, 7, 10, 11], including productivity of a user's primary work (work to be interrupted) or quality of decision making, negative effect on a user's emotional states and social attribution [4], and even negative effect on a user's psycho-physiological states [10].

Although a user, of course, can configure the notification system and can even disable the notification delivery, such operation negates the benefits of the notification system and cannot fulfill the needs of the users for the speedy and timely delivery of new information. Users prefers to keep notification systems on, even given the interruption costs, rather than simply disabling the system and check the new information manually, according to the previous research [12].

2.1. Existing Research on Mitigating the Cost of Notifications

Past research for the interruption overload problem can be categorized into two different approaches, (a) scheduling (deferring) notifications, and (b) mitigation of notifications.

A "breakpoint" is often used as a target timing of deferred notification to users, in the first "deferring" approach. Breakpoint [8] is a concept found in psychology field. A boundary between two contiguous discrete actions, when a human's perceptional system segments human activities into a series of hierarchically-structured actions, is called a breakpoint. Iqbal et al [13] found that there are at least three different granularities of breakpoints (Fine, Medium, and Coarse), that users can reliably detect during interactive computing tasks. Other research [4, 14, 15] has also observed interruption cost, including resumption lag of the user's primary task and subjective frustration value, when notifications are deferred until the breakpoint timings.

The other mitigation approach, instead, changes modality of the notification delivery, such as use of vibration instead of sound, while keeping the timing of notifications to the original timings, in order to reduce a user's cognitive load. This approach contributes to lowering the saliency of the notifications.

This paper particularly focuses on the deferral of the notifications, although the two approaches introduced above can be complementary with each other and not mutually exclusive. Given our research background where users have been facing an increasing number of notifications, our research interest is on the notification timing which would seem to have greater potential impact on user's interruption overload than saliency.

In contrast to existing research that focused on notification deferral in desktop computing area, mainly with a single computing device and with an in-lab controlled evaluation, notification deferral (1) in mobile devices, (2) in real-time, (3) with diverse types of applications, and (4) with in-the-wild real world evaluation, are our main and important research opportunities.

3. Adaptive Notification Scheduling on Smart Phones

In this section, we clarify the requirements for adaptive notification scheduling on smart phones, along with several recent trends in notification, in order to scope our research contribution.

3.1. Recent Trends of Notifications

Reflecting the recent trends in ubiquitous computing described in Section 1, we point out and focus on the following distinctive characteristics of notifications in such environments.

- **Increasing diversity in types and sources of notifications:** Based on an increasing number of applications and services, communication channels and connections between users, there have been an increasing diversity of types and sources of notifications, including updates from social networks, signals from sensors, and queries from participatory sensing systems [16],
- **Multiple mobile devices as targets:** Users are carrying multiple mobile (and wearable) devices, including smartphones, tablets, smart glasses and smart watches [2, 1], all of which can be targets of notifications.
- **Wider range of urgency level:** While most notifications are informative in nature, some require almost instant reaction: e.g., Early Earthquake Warning (EEW) [17] notifications for which users need to physically react within a few seconds.
- **Increasing length of interruptive periods:** Recent lifestyles include always having access to one's mobile devices, making interruption overload an issue all day long, even while a user is sleeping.

3.2. Principles for Attention Status Sensing

To defer notifications to user's breakpoints, we must be able to sense their attention state. Based on the previous literature, we denote the following as principles in attention status sensing.

- **Feasibility for mobile devices:** Users carry and use mobile devices, such as smart phones or tablets, for everyday computing and communication. Thus a breakpoint detection system needs to work on a mobile platform, in terms of energy-efficiency, available sensors, etc.

- **Real-time sensing:** To support notification adaptation and deferral on the fly, the sensing needs to be performed in real-time.
- **Applicability to diverse types of notification sources:** The breakpoint detection system needs to work for diverse types of notification sources.
- **All-day-long use:** Breakpoint sensing needs to be performed all day long, or at least as long as the user's notification system is available.

4. Design of Attelia

In this section, we present our design of Attelia, based on the requirements we described in the previous section. Attelia detects appropriate timings for delivering notifications to users, with three distinctive features. First, it detects those timings on smartphones, without the use of an external server or any psycho-physiological sensors. Second, Attelia detects breakpoints in real-time (not post-hoc) so that it can be used to adapt notification timings at run-time. Finally, the breakpoint detection can be applied to a wide range of applications installed on users smartphones.

The following subsections describe our approach for performing breakpoint detection that satisfy these three features, including: (1) using breakpoints to temporally target interruptions, (2) using application usage as a sensor, and (3) using machine learning to perform real-time breakpoint detection.

Since our research focus is on the user's "mobile experience" during his/her active manipulation of devices, Attelia scopes breakpoint detection during their active engagement with mobile devices, and does not consider moments when users are not interacting with them.

4.1. "breakpoint" as a Temporal Target for Interruption

Related work in real-time sensing of available user attention or cognitive load shows that at least two psycho-physiological sensors are needed even in non-mobile situations [18]. Given the burden of wearing a psycho-physiological device constantly, our approach only uses the users' mobile devices, and attempts to sense more coarse-grained, but easier to sense signals, from which appropriate timings for notifications can be inferred.

4.2. Application Usage as a Sensor

With our scoping to active use of mobile devices, we focus on how users interact with mobile applications and use that information to detect a user's breakpoints. We focus on application usage and not physical sensors, despite their wide proliferation on mobile devices for two reasons: simplicity of implementation and reducing the reliance on a sensor that may not exist on all target mobile devices (or may be mounted in different locations).

Table 1 shows some possible knowledge sources for identifying breakpoint and and Table 2 shows, for each source type, how it can be acquired. The application-related knowledge and information can include both relatively static knowledge that is specific to each application, such as when users transition between multiple "stages" in game applications, and that are designed and implemented by the application developers in the development phase; and relatively dynamic information, such as run-time status and events that result from the running applications. Using knowledge from the internals of any specific application is not feasible given the huge number of applications available and the fact that application developers would need to expose internal information at development time. Instead, we collect run-time status events from the operating system and executing applications, and use them to identify relationships to ground truth values of interruptive overload provided by users, during a training phase.

4.3. Real-Time Detection with Machine Learning techniques

Similar to previous work on activity recognition, our approach also uses machine learning-based classification to understand these relationships. For each time frame T_f, a feature vector V is extracted from the sensed data, and a trained classifier identifies the time frame as a user breakpoint or not.

Table 1: Approaches of Knowledge Collection for Breakpoint Detection

Approaches on Knowledge Source of Breakpoint	Examples of Data Types
Application-specific breakpoint knowledge	explicit breakpoint declaration inside application, explicit future breakpoint forecast inside application
Runtime status/event of systems and applications	stack trace, number of threads, thread names, memory consumption Android API invocation, system call invocation, rendered screen image, Low-level GUI events, switches between applications

Table 2: Timings of Knowledge Input and Data Collection

Approaches on Knowledge Source of Breakpoint	Knowledge on Breakpoints: When? By Who? and How?		Data Collection at Application Run-Time
	Application Development Phase	System Training Phase	
Application-specific breakpoint knowledge	Embedding additional API calls to provide explicit breakpoint knowledge (by application developer)	None	From API calls embedded inside running applications
Runtime status/event of systems and applications	None	Ground truth annotation of collected status/event information (by application users)	From the middleware and operating system

5. Attelia System Architecture

Figure 1 shows the system structure of Attelia implemented on the Android 4 platform. Attelia consists of an Android service that includes several internal components for UI event logging, breakpoint ground truth annotation logging, as well as a machine learning engine that performs feature extraction and classification (using an embedded Weka [19] engine).

5.1. Execution Modes

Attelia can execute in ground truth annotation mode, off-line training mode or real-time breakpoint detection mode. In the annotation and detection modes,

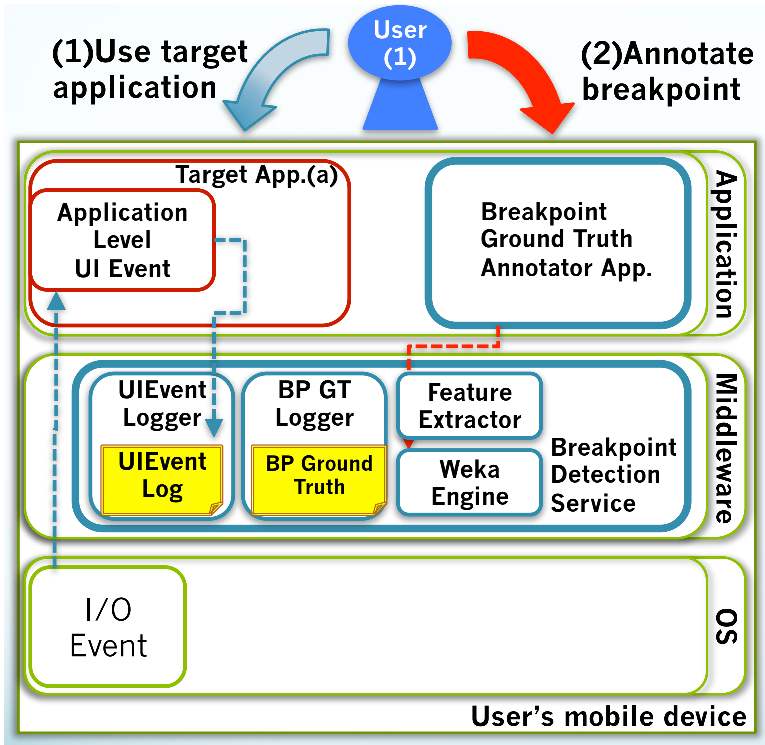

Figure 1: System Architecture of Attelia on Android Platform

the UIEventLogger component listens to the stream of incoming UI events and records relevant events to the log file.

- **Ground truth collection:** In this mode, users manually provide ground truth about breakpoints during application usage. Figure 2 shows a screen-shot of Attelia, with our Annotation widget floating on the screen. While manipulating ordinary Android applications, users push the floating button when they are switching activities. The Attelia service records the stream of UI events (excluding those from the annotation button) and breakpoint timestamps (moments when the annotation button was pushed).

- **Off-line model training:** In this mode, feature extraction and classifier training is executed off-line, using the previously-stored sensor and ground

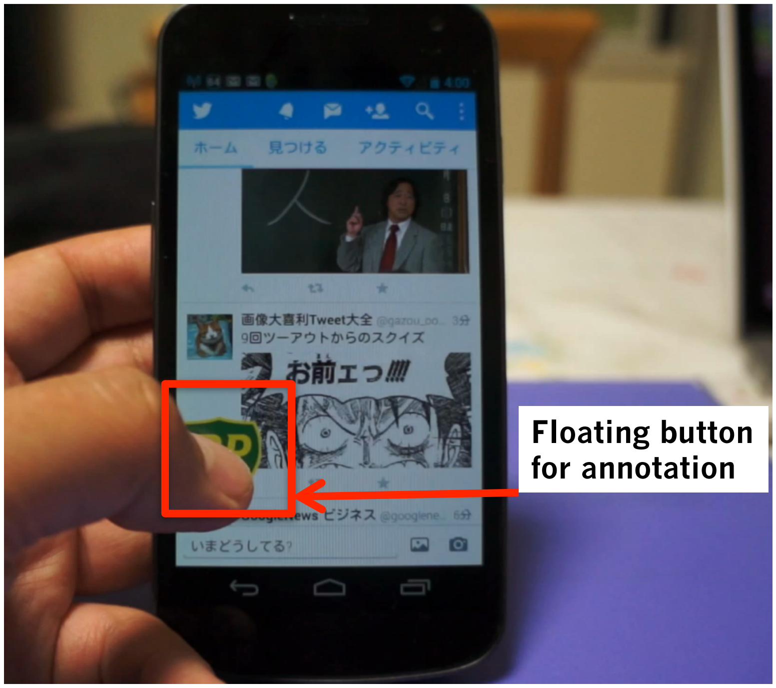

Figure 2: Ground Truth Annotation with Attelia

truth data.

- **Real-time mobile breakpoint detection:** Sensing, feature extraction, and classification with a previously-trained model is performed in real-time on a smartphone.

5.2. Sensing Data and Features

To obtain the stream of UI events from the middleware, we use the Android Accessibility Framework. Using this framework, Attelia can collect UI events and data about the UI components the user is interacting with. A list of the UI Events we collect is shown in Table 3.

From these events, we extract the 45 features outlined in Table 4. These features are extracted for each "time frame" and stored during ground truth annotation, and are fed to the Weka machine learning system for classification

Table 3: UIEvent Collected in Attelia

Event Types	Events
View	`View clicked`, `View long clicked`, `View selected`, `View focused`, `View text changed`, `View selection changed`, `View text traversed at movement granularity`, `View scrolled`
Transition	`Window state changed`, `Window content changed`
Notification	`Notification state changed`

Table 4: Features used in Attelia

Feature Types	Features
Rate of occurrence of each UI Event type inside the frame	snipped (one for each event type presented in Table 3)
Statistics on the status of the event source UI component	*rate(*`isEnabled`*)*, *rate(*`isChecked`*)*, *rate(*`isPassword`*)*
Statistics on the events' timings in the frame	*min_timegap*, *mean_timegap*, *max_timegap*, *stdev_timegap*
Statistics on the location of the event source UI components	*min.*, *mean.*, *max.*, *stdev.*, *the value of the smallest rectangle*, *the value of the biggest rectangle* of `X-left`, `X-right`, `X-width`, `Y-top`, `Y-bottom`, `Y-height`

during breakpoint detection. We attempted to be exhaustive in providing possible features to capture as many characteristics of the real execution environment as possible.

5.3. Frame Length

With an expectation that our choice of time frame length T_f will affect our ability to perform breakpoint detection, we conducted a small user study to investigate the impact of frame length on detection accuracy. Eight participants were recruited: university undergraduate and graduate students and staff with ages between 18 and 27 years, who use smartphones daily. Each participant manipulated five common Android applications (Twitter, Yahoo News, YouTube, Kindle, Browser) for 5 minutes each (per application) performing everyday tasks, and indicated their breakpoints using our floating annotator button. Participants used a Samsung Galaxy Nexus [20] smartphone running Android version 4 for the experiment.

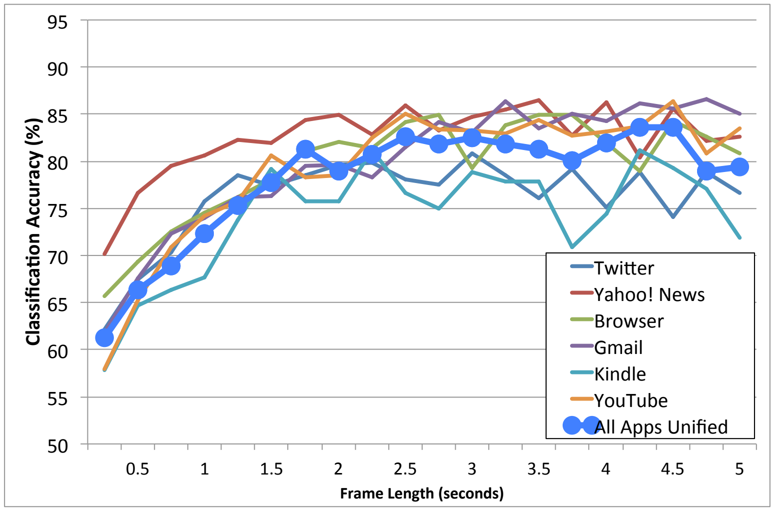

Figure 3: Classification Accuracy and Frame Length

Figure 3 shows the classification accuracy results with different frame lengths (0.25 to 5 seconds), using 10-fold cross validation on Weka 3.7.9 and J48 classifier. The data for each application is aggregated from all eight participants, and is represented as a separate line in the graph. An additional line in the graph (bolded) represents all application data aggregated together from all the participants. Accuracy is low when the frame length is very short (e.g., 0.25 seconds), because there are not enough sensed UI events within that time span to achieve a high classification accuracy. However, around 2 to 2.5 seconds, the accuracy begins to stabilize. At the 2.5-second setting, accuracy was 82.6%, precision was 82.7% and recall was 82.3%.

5.4. Power Saving

To save power, we disable real-time feature extraction and classification when the device screen is off, as we are concerned with detecting breakpoints when the user is engaged with the device. In addition, if no UI event occurs within a given time frame, no classification is performed.

Table 5 shows a power comparison between using our UI events and using common sensors. We used a Samsung Galaxy Nexus with Android 4.4.4 and

Table 5: Comparisons of Power Consumption Overhead

Sensor Type	Frequency (Hz)	Overhead (mW)
UI Events	10	51.70
Accelerometer	120	102.90
	60	48.76
	15	12.08
Gyroscope	100	158.88
	50	129.24
	15	74.04

measured the data with a Monsoon Power Monitor [21]. Each table value is the average of five 5-minute measurements. The result shows that the overhead of our UIEvent data collection software is quite low compared with other sensors and considering that multiple types of sensors, such as the accelerometer, gyroscope and GPS, are used in combination for many activity recognition systems,

In Attelia, since the number of incoming UI events depends on user interaction, we looked to our user study data to determine an appropriate number. Based on the data collected from 30 users for 16 days, the average number of UI events was 10.6 per second on average ($min = 1$, $max = 549$, $stdev. = 15.1$) during users' active manipulation of their device. We then logged the power consumption using Android instrumentation that fired approximately 10 UI events every second. To compare to the other sensors, we implemented a basic application which reads and stores the sensor data with the specified frequency.

5.5. Portable Implementation

Attelia is implemented as a "Service" inside the Android platform. By appropriately setting the permissions for the service, it can log the stream of UI events, such as tapping, clicking, and scrolling or modifications of UI components inside the currently-active Android application without requiring root privileges. This implementation allows the service to be distributed through the Google Play store and contributes to the deployability of the system to end users.

6. Controlled User Study

To further understand how Attelia works, we conducted a controlled user study based on our implementation. The overall purpose of study was to investigate if providing notifications to users at the timing of breakpoints detected in real-time lowers user's cognitive load.

6.1. Participants

For the study, 37 participants were recruited. Among them there were university students, staff members, and research engineers, with ages between 19 and 54. All the participants were smart phone users in their daily lives. Subjects were not told the specific objectives of the study at the beginning, and not paid for the participation.

6.2. Experimental Setup

For the study, we prepared Samsung Galaxy Nexus smart phones with Android 4.3 for each participant. The original notification feature of each phone was disabled. For our experiment, we installed our Attelia prototype software and six representative Android applications (Twitter, Gmail, Yahoo News, YouTube, Kindle, and Browser) in to each phone. The Attelia service was configured to "real-time mobile breakpoint detection" mode, with a J48 decision tree classifier trained through our previous experiment, with 2.5-second time frame T_f setting.

We prepared four different notification strategies in this study, namely (1) disabled (no notification at all), (2) random timing (emulating a "conventional" notification situation), (3) breakpoint timing (our approach), and (4) non-breakpoint timing (interrupting at times that our system determines as inopportune). The approaches (2), (3), and (4) were configured to have intervals of at least 30 seconds between two consecutive interruptions. During the study, each participant was exposed to one of the four different notification strategies for being interrupted by notifications. Strategies were changed for each participant, and for each session. The order of the selection of the strategies was randomized, and the information on the selection was not revealed to the participants.

On the interruptive tasks, a full screen pop-up window appeared on the screen to ensure that the interruption would not go unnoticed, when participants were interrupted. The pop-up contained the first paragraph from a random news article. During interruption, the participants were given a interruptive task: To read the paragraph and select an appropriate title for the article given three options. We chose this interruptive task from the similar previous interruption studies [22, 23]. Subjects were asked to finished the task as fast and accurately as possible. After the participant finished the task, the pop-up window disappeared so that the user could return to the original task that she was performing.

6.3. Experiment Procedure

Our experimental procedure contained two parts. In the first part, each user was given a printed email and was told to compose and send an email with the specified text using the Gmail app. Each user repeated this task five times, with different text and different notification strategy. In the second part, each user was asked to use each of the other selected applications (Twitter, Yahoo News, YouTube, Kindle, Browser) as they "normally would" for 5 minutes each, and experienced a different notification strategy with each application.

The order of the email texts (part 1), applications (part 2), and notification strategies were counterbalanced using a balanced Latin Square to remove ordering effects. Since there were 4 strategies, and the email and app use tasks were performed 5 times, each user saw one strategy twice, which was randomly selected. A repeated measures within-subject design was used with the notification strategy as factors.

6.4. Measurements

To measure participant's subjective cognitive load, we used the web page version of the NASA-TLX[24] questionnaire. Each participant was asked to answer the questionnaire after each task (i.e., a total of 10 times per participant).

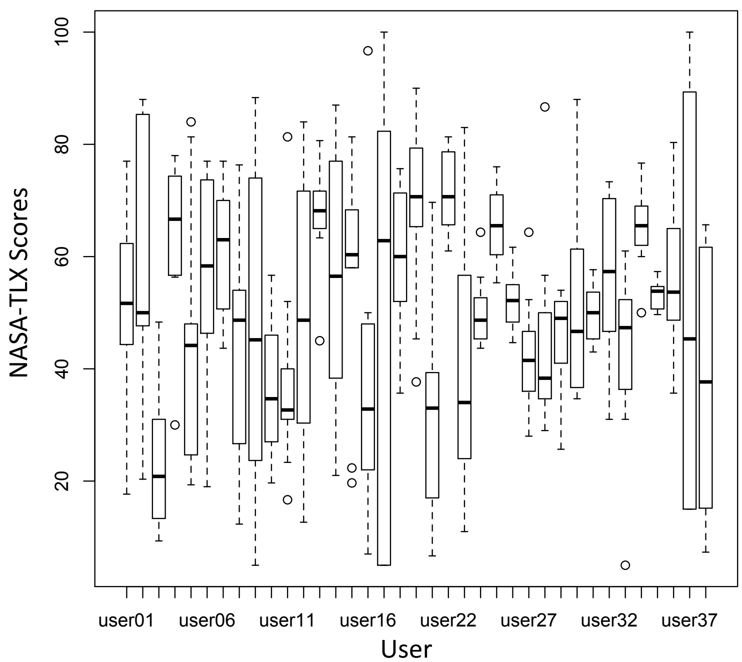

Figure 4: Variance of NASA-TLX WWL Scores (Controlled User Study)

6.5. Result Analysis: Subjective Cognitive Load

As shown in Figure 4, we observed differences in the range of subjective cognitive load (NASA-TLX Weighted Workload (WWL) score) in terms of their "individual" means and variances across different notification strategies. More specifically, we noticed that some of our participants were more sensitive (higher variance in their WWL) to the different notification strategies. Also some of our participants seem to not react (e.g., insensitive) to the notification strategies (low variance in their WWL). This fact motivated us to try to identify clusters within our user population.

First, we conducted hierarchical clustering (using the Ward method and Euclidean distance) on the variance of each participants NASA-TLX WWL scores, in order to observe the dissimilarity between users. Figure 5 shows the resulted dendrogram.

Based on the height between two clusters in the dendrogram figure with the

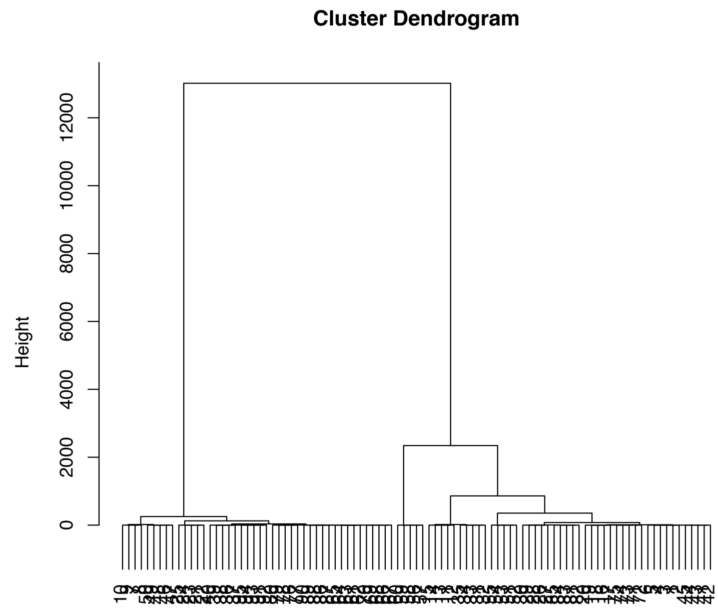

Figure 5: Dendrogram from Structured Clustering (Personal WWL score variances) (Controlled Study

Table 6: Two Clusters in the Controlled User Study

Cluster name	Users	Mean WWL Stdev.
"WWL-sensitive users"	19	23.11
"WWL-insensitive users"	18	9.92

used Ward method, we identified 2 distinct clusters. The number of participants and the mean of personal WWL score standard deviation in each cluster are shown in Table 6. We named these cluster "WWL-sensitive users" (those with higher score variance among the different strategies) and "WWL-insensitive users", since this clustering was on the variance of each participant's "personal score variance".

Figure 6 shows the average NASA-TLX WWL scores for the different notification strategies, for the two clusters respectively.

The most significant finding in this analysis is that, for the "WWL-sensitive

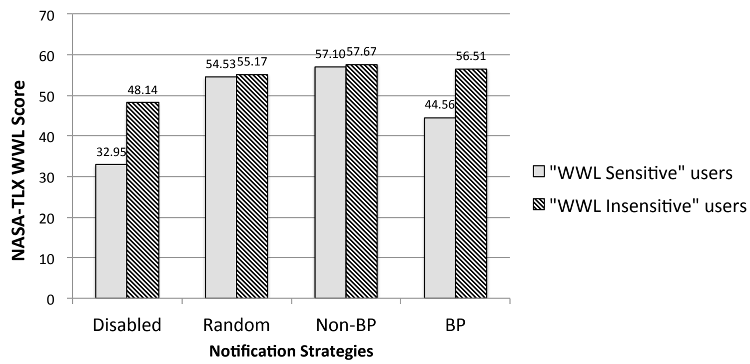

Figure 6: NASA-TLX WWL Scores for Each Cluster (Controlled User Study)

users", a 46% decrease in cognitive load was observed in our breakpoint strategy ("BP") results, compared to the cognitive load in the random strategy ("Random"), that emulates how people are currently experiencing interruptions on the standard Android notification system. "BP" strategy (cognitive load score of 44.56) resulted in only an increase of 35% in cognitive load when compared to the baseline "Disabled" strategy (cognitive load score of 32.95) with no notifications, while "random" strategy (cognitive load score of 54.53) resulted in an increase of 66%. Also, as we expected, the non-breakpoint strategy ("Non-BP"), where notifications were be intentionally displayed only at the timings that were not detected as breakpoints, resulted in the highest 73% increase in cognitive load, with a score of 57.10.

A Friedman test revealed a significant effect of notification strategy on the WWL score ($\chi^2(3) = 16.5, p < 0.05$). A post-hoc pair-wise comparison using Wilcoxon rank sum tests with Holm correction showed the significant differences between "Disabled" and "Random" ($p < 0.01$, $\gamma = 0.34$), between "Disabled" and "non-BP" ($p < 0.01$, $\gamma = 0.39$), between "Disabled" and "BP" ($p < 0.05$, $\gamma = 0.29$), between "Random" and "BP" ($p < 0.05$, $\gamma = 0.24$), and between "non-BP" and "BP" ($p < 0.05$, $\gamma = 0.26$). Between "Random" and "non-BP", a statistical difference was not observed.

On the other hand, For the "WWL-insensitive users", the result shows the insensitivity of the participants. As expected, from our Friedman test and pair-wise test with Wilcoxon rank sum tests, no significant differences were observed during “Random”, “BP”, and “non-BP”, while significant differences between “Disabled” and the other strategies were found (Friedman test with the effect of notification strategy on the WWL score ($\chi^2(3) = 9.4, p < 0.05$)). The significant differences from the post-hoc test using Wilcoxon rank sum tests with Holm correction are observed between “Disabled” and “Random” ($p < 0.01$, $\gamma = 0.30$), between “Disabled” and “non-BP” ($p < 0.01$, $\gamma = 0.35$), and between “Disabled” and “BP” ($p < 0.01$, $\gamma = 0.34$).

7. In-the-Wild User Study

Based on the promising results from our controlled user study, we proceeded to “in-the-wild” user study to better understand how Attelia could reduce user’s cognitive load in the user’s real computing lives. In this study, we installed our Attelia service on each participant’s smart phone. We compared multiple different notification strategies and investigated if notifications displayed at the timings of detected breakpoints could reduce participants’ cognitive load.

7.1. Participants

For this study, 30 (20 male and 10 female) people, who are using “Android 4.3 (or above) smart phone” in their daily lives, were recruited as the participants. Among the participants there were university staff members and students, with ages ranging from 18 to 29 years old. 20 participants belonged to computer science and information technology related departments, while the remaining participants belonged to other schools, such as social sciences, economics, and psychology. All of the participants were using Android OS version 4.3 (or above) smart phones in their daily lives. Subjects were paid $60 for their participation.

7.2. Experimental Setup

We packaged the Attelia service and some additional experiment-related data collection services and their parameters into a single Android “service”. With each participant's permission, we installed the service to each participant's own smart phone. The Attelia service was configured to "real-time mobile breakpoint detection" mode, with a J48 decision tree classifier trained using our previous experiment, with a 2.5-second time frame T_f setting. In this study, we prepared three different notification strategies, namely (1) "Disabled" (no notification), (2) "Random", and (3) "Breakpoint" (our approach). Everyday, for each user, the data collection logic randomly chose one of these strategies to be used for notification throughout the day.

We set the following study-specific parameters for each user: (1)the daily maximum number of interruptive tasks to 12, (2)the minimum interval between two consecutive notifications was set to 15 minutes, (3)the maximum interval was set to 30 minutes, (4)the service was configured to show notifications only from 8AM to 9PM daily. These parameter values were carefully chosen to get enough data points without requiring too much effort from the participants. The last was estimated from interviews to the participants about their daily life patterns.

Regarding on the interruptive task, a full screen pop-up window appeared on the screen to ensure that the interruption would not go unnoticed, when participants were interrupted. The first screen asked if the timing was during a natural breakpoint. The second pop-up was shown regardless of the user's answer to the question. The second screen presented the same interruptive task prepared for the controlled study. The participants were asked to finish the task as fast and accurately as possible. After the participants finished the task, the pop-up window disappeared so that they could go back to the original task.

7.3. Experiment Procedure

Our experimental procedure consists of the following three parts. (1) Each participant had a meeting with a study researcher at the beginning of the user study. The participant received basic information and instructions on the study, followed by signing a consent form. Afterwards, the researcher installed and started the Attelia software on the participant's smart phone. The existence of multiple different notification strategies was explained to the participants, but the detailed behavior was not explained. (2) The 16 day long experiment started after the meeting. As mentioned above, every day the notification strategy for each user was randomly changed. Information about the notification strategy working every day was not revealed to the participant. During the experiment, at the end of each day, a NASA-TLX survey was sent to all participants. Each participant was required to individually answer NASA-TLX survey every night, for 16 days. (3) After the 16-day period finished, participants filled out the post-experiment survey, uninstalled the Attelia service, and were paid.

7.4. Measurements

The Attelia service recorded the time taken to respond to the first and second notifications, time to answer the quiz, and the answer to the quiz. The data was uploaded to our server every night. The NASA-TLX questionnaires (implemented as a web page on our web server) were sent to each user via email every night, thus the survey results were stored inside our database on the server.

7.5. Result Analysis: Subjective Cognitive Load

From the experiment, we collected the answers to NASA-TLX surveys from each of 30 participants over 16 days. The data from 3 users was discarded due to several issues: data not properly recorded and uploaded to the server or the user forgot to fill out the daily survey. Our final data set consisted of 27 users' data and we used it for the following data analysis.

As shown in Figure 7, again, we observed differences in the range of subjective NASA-TLX WWL score in terms of individual personal means and variances across different notification strategies. More specifically, we observed once again sensitive and insensitive (to the notification strategy) users.

Thus, we first conducted a hierarchical clustering with the Ward method and Euclidean distance on the variance of each users NASA-TLX WWL scores. Figure 8 shows the resulting dendrogram for this clustering.

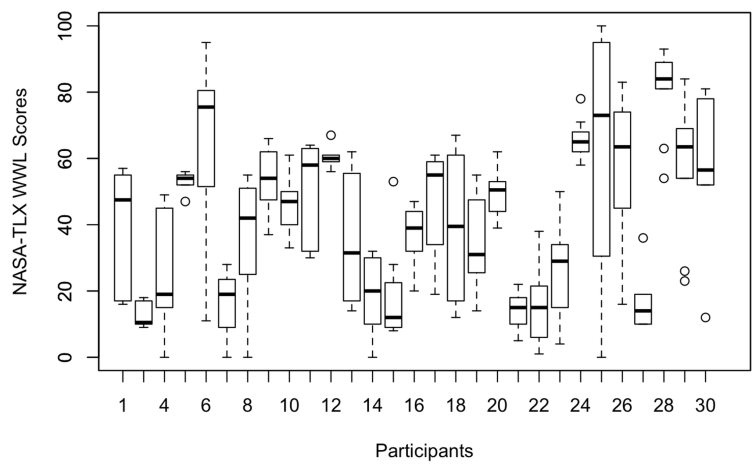

Figure 7: Variance of NASA-TLX WWL Scores (In-the-wild User Study)

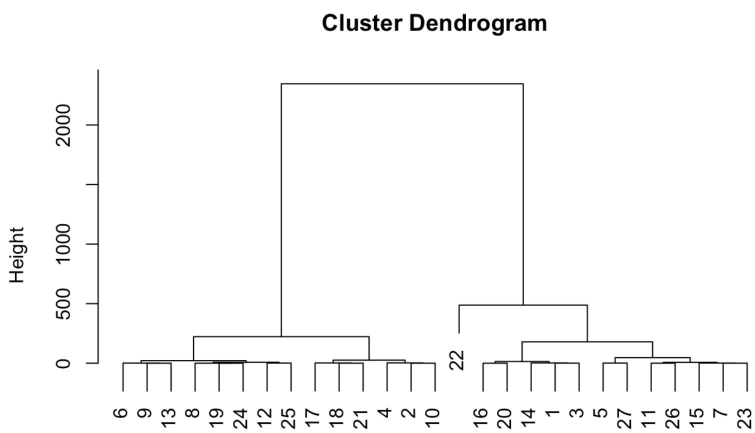

Figure 8: Dendrogram from Structured Clustering (Personal WWL score variances) (In-the-wild Study

Based on the height between two clusters in the dendrogram figure with the Ward method, we again identified 2 distinct clusters. Table 7 shows the the number of users and the mean of personal WWL score standard deviation in each cluster. Following our naming convention in the controlled study, we named the clusters "WWL-sensitive users" and "WWL-insensitive users" respectively.

The average NASA-TLX WWL scores, for each notification strategies and

Table 7: Two Clusters in the Wild User Study

Cluster name	Users	Mean WWL Stdev.
"WWL-sensitive users"	13	21.38
"WWL-insensitive users"	14	8.19

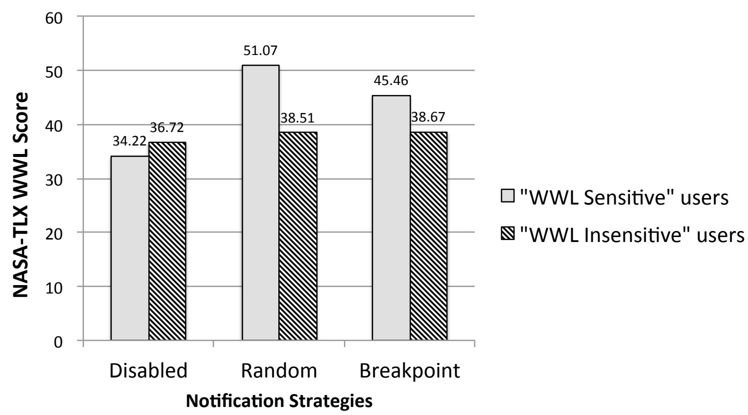

Figure 9: NASA-TLX WWL Scores for Each Cluster (In-the-wild User Study)

for each cluster, are illustrated in Figure 9.

For the "WWL-sensitive users", the results shows the same trend as we observed in the controlled user study. A 33% decrease in cognitive load was observed in our breakpoint strategy ("Breakpoint") results, compared to the cognitive load in the random strategy ("Random"), that emulates how people are currently experiencing interruptions on the standard Android notification system. "Breakpoint" strategy (cognitive load score of 45.46) resulted in only an increase of 33% in cognitive load when compared to the baseline "Disabled" strategy (cognitive load score of 34.22) with no notifications, while "Random" strategy (cognitive load score of 51.07) resulted in an increase of 49%.

A Friedman test revealed a significant effect of notification strategy on the WWL score ($\chi^2(2) = 8.5$, $p < 0.05$). A post-hoc pair-wise comparison using Wilcoxon rank sum tests with Holm correction showed significant differences between "Disabled" and "Random" ($p < 0.01$, $\gamma = 0.37$) and between "Random"

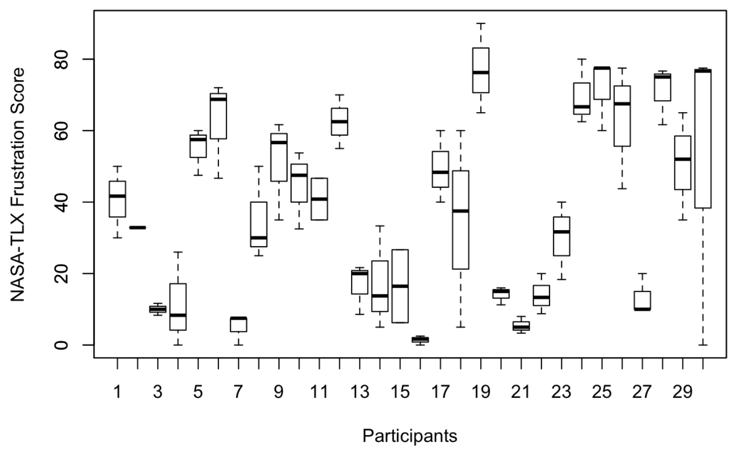

Figure 10: Frustration score variances among participants

and “Breakpoint” ($p < 0.05$, $\gamma = 0.20$),

For “WWL-insensitive users”, on the other hand, our Friedman test and pair-wise test with Wilcoxon rank sum tests showed no significant differences between all of three strategies.

7.6. Result Analysis: Subjective Frustration

We also conducted another analysis on user’s subjective frustration value collected in daily NASA-TLX surveys, since the frustration value looks the key element among 6 different elements in the survey. Figure 10 shows the variances of the frustration scores for all participants.

Similarly, for the frustration scores, since differences in the variance among users were observed, we first conducted a hierarchical clustering using the Ward method and Euclidean distance on the variance of each users NASA-TLX frustration scores. The resulted dendrogram is shown in Figure 11. Similar to the analysis on the WWL score variances, we observe a separation between the observed 2 clusters, thus concluded the size of the clusters to 2.

Table 8 shows the number of users and the mean personal frustration score

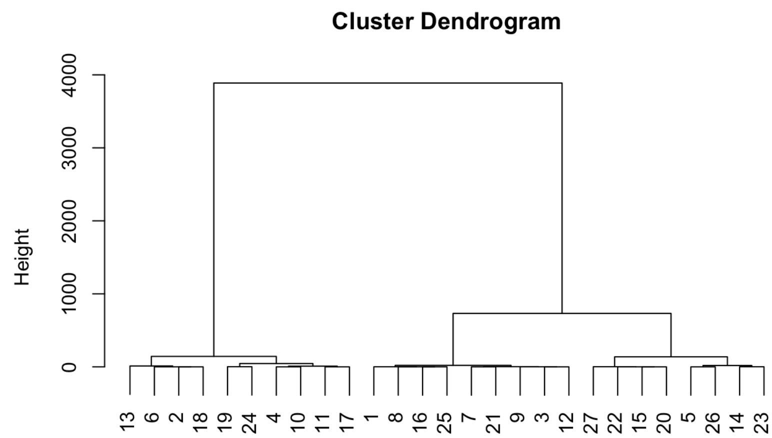

Figure 11: Dendrogram from Structured Clustering (Personal frustration score variances) (In-the-wild Study

Table 8: Two Clusters on the Frustration Score Variances

Cluster name	Users	Mean Frustration Stdev.
“FRU-sensitive users”	17	25.29
“FRU-insensitive users”	10	7.72

Table 9: Comparisons between Two Clustering Analysis

	FRU-sensitive users	FRU-insensitive users
WWL-sensitive users	13	0
WWL-insensitive users	4	10

standard deviation in each cluster respectively. Also, Table 9 shows the comparisons between the WWL-based clustering and the frustration-based clustering. All of 13 participants in the “WWL-sensitive users” cluster are clustered in to “FRU-sensitive users” cluster. On the other hand, 10 out of 14 users in “WWL-insensitive users” are clustered in to “FRU-insensitive users” while other 4 users are clustered in to “FRU-sensitive users”.

Figure 12 shows the average frustration scores for the different notification strategies, for the two clusters respectively.

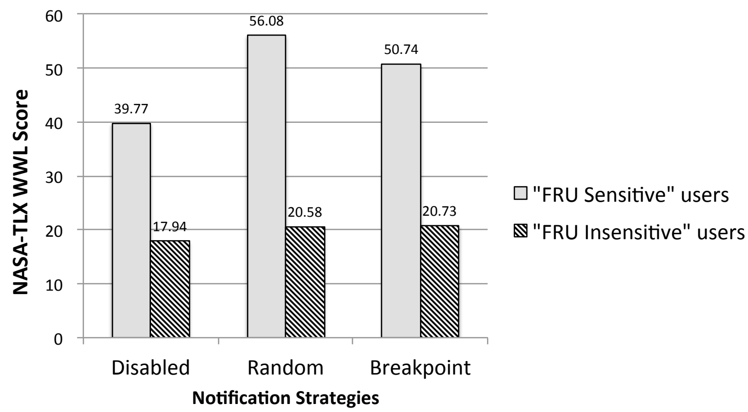

Figure 12: Frustration Scores for Each Cluster (In-the-wild User Study)

For the “FRU-sensitive cluster”, we observe the same trend as we saw in our WWL score analysis. A 33% decrease in frustration was observed in our breakpoint strategy (“Breakpoint”) results, compared to the cognitive load in the random strategy (“Random”). “Breakpoint” strategy (frustration score of 50.74) resulted in only an increase of 27% in cognitive load when compared to the baseline “Disabled” strategy (cognitive load score of 39.77) with no notifications, while “Random” strategy (cognitive load score of 50.74) resulted in an increase of 41%. A Friedman test revealed a significant effect of notification strategy on the WWL score ($\chi^2(2) = 4.7$, $p < 0.05$). A post-hoc pair-wise comparison using Wilcoxon rank sum tests with Holm correction showed significant differences between “Disabled” and “Random” ($p < 0.05$, $\gamma = 0.33$), between “Disabled” and “Breakpoint” ($p < 0.05$, $\gamma = 0.22$), and between “Random” and “Breakpoint” ($p < 0.05$, $\gamma = 0.17$).

On the other hand, for the “FRU-insensitive users”, no significant differences were observed between all of three strategies, by our Friedman test and pair-wise test with Wilcoxon rank sum tests.

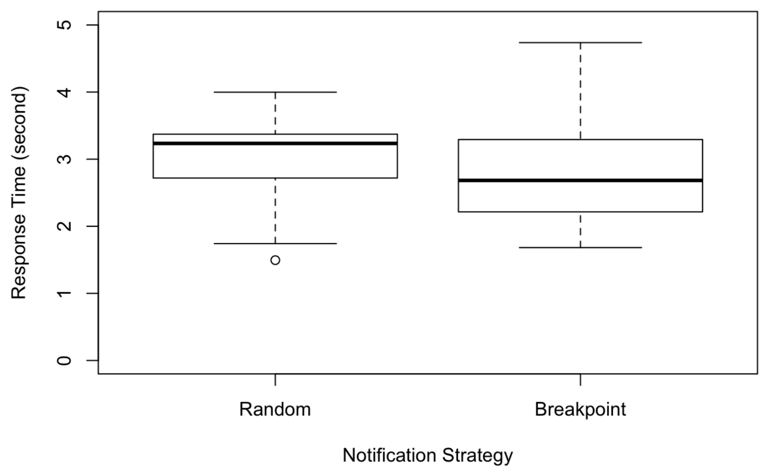

Figure 13: Response Time to the First Pop-up

7.7. Result Analysis: Response Time for the First Pop-up

Figure 13 shows our next analysis on the response time to the first pop-up. The response time is the time differences from when the first pop-up is shown on the smart phone screen to when it was answered by the user. After the user study, we obtained 1130 data points for the "Random" strategy and 1032 data points for the "Breakpoint" strategy. The average response time was 3.18 seconds in "Random" and 2.77 seconds in "Breakpoint" respectively. Our Wilcoxon Signed-rank test showed that there is a significant effect of strategy ($W = 343$, $Z = -3.19$, $p < 0.05$, $\gamma = 0.37$).

7.8. Result Analysis: Response Time for the Second Pop-up

Next we analyzed response time for the corresponding second pop-up questions. Again, the response time is the time differences from when the second pop-up is shown on the screen to when it was answered by the user. Figure 14 shows the results. The average response time in "Random" strategy is 5.97 seconds while the average response time in "Breakpoint" strategy is 5.88 seconds. Our Wilcoxon Signed-rank test did not showed significant difference the response time values of the strategies. Also, the same types of tests combined

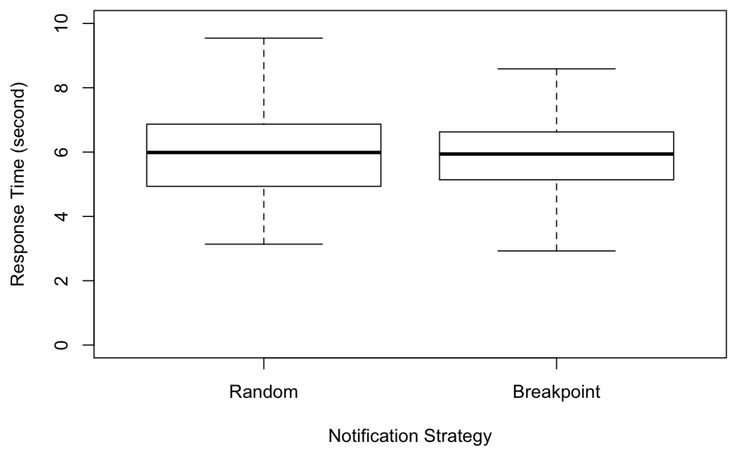

Figure 14: Response Time to the Second Pop-up

with the clustering (either WWL or frustration scores) did not show any significant difference.

From this analysis result, our hypothesis is that, since the target of user's attention was already switched from the user's primary task to the interruption at the timing of the first pop-up, regardless of the type of the notification strategies used, the response time values for the second pop-up are not significantly different between the notification strategies (of the first pop-up).

7.9. Result Analysis: Correct Answer Rate for the Second Pop-up

Another analysis on the second pop-up was on the correct answer rate for the second pop-up screen. Figure 15 shows the results. The correct answer rate is 87.0% in "Random" strategy and 87.8% in "Breakpoint" strategy. However, our Wilcoxon Signed-rank test did not showed significant difference the response time values of the strategies. Furthermore, the same types of tests combined with the clustering (either WWL or frustration scores) did not show any significant difference. This analysis result supports our hypothesis on the target of user's attention mentioned above.

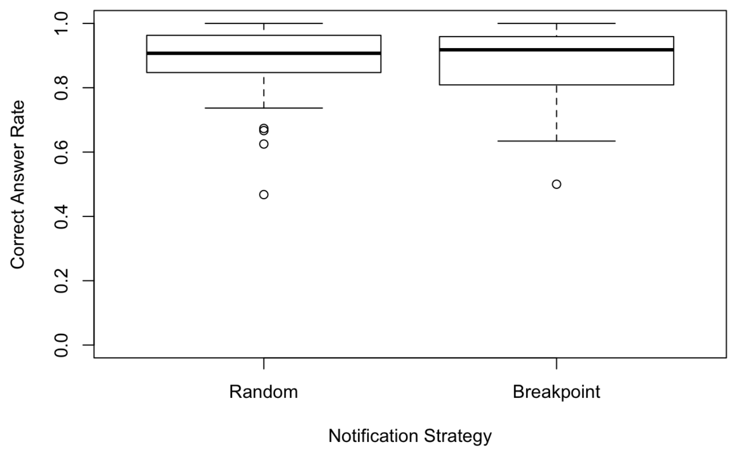

Figure 15: Correct Answer Rate in the Second Pop-up

7.10. Post-Experiment Survey

After the 16 day experiment has finished, we conducted an instant post-experiment survey for each user, at the end of the user study. Table 10 and 11 summarizes the results. We asked each participant following 5 questions.

1. Did you realize any differences between several different notification strategies that changed daily, excluding "No notification" strategy?
2. Do you think the differences in Q(1) affected your mental workload?
3. Do you think the differences in Q(1) affected your response time to answer the quiz?
4. With our software, did you see any difference in your phone's battery life?
5. With our software, did you observe any performance degradation in your phone?

For questions 1 to 3, we asked the participants to answer each question by using 5-level likert scale (1 – Strong disagree, 2 – Disagree, 3 – Neutral, 4 – Agree, and 5 – Strongly agree). For questions 4 and 5, we asked the participants to answer each question by using another 5-likert scale (1 – Not at all influential, 2 –

Slightly influential, 3 – Somewhat influential, 4 – Very influential, 5 – Extremely influential).

Table 10 summarizes the answers of question 1 to 3. For the question (1) "Did you realize any differences between several different notification strategies that changed daily, excluding "No notification" strategy?", the answer with the biggest number of the participants was "Disagree", while the answer with the second biggest number of the participants was "Agree". From this result, we hypothesize that this answer may be related to the clusters we observed in our NASA-TLX score analysis. However, we could not further analyze these survey answers in terms of possible matching against the cluster we previously generated, since there were several data inconsistency in the user ID field of the survey answer data. At the survey, we asked each participant to input her/his own user ID manually. However, we eventually found that several participants have input the wrong user ID number, then it was not possible to analyze the data, referring their user ID values.

For the question (2) "Do you think the differences in Q(1) affected your mental workload?", more than half of the participants (16 out of 30) agrees or strongly agree, while 7 participants disagrees or strongly disagree. The same trend was observed for the question (3) "Do you think the differences in Q(1) affected your response time to answer the quiz?". 18 participants agrees or strongly agree to the question, while 5 participants disagrees or strongly disagree. Again, we could not analyze the results further on the relationships with the observed clusters, due to the inconsistent data on the user IDs.

Table 11 summarizes the answers of question 4 and 5. For the question (4) "With our software, did you see any difference in your phone's battery life?", 10 out of 30 participants answered that it was not influential at all. Although 20 participants were aware of some level of change in power consumption, the total number of users who answered "Not at all influential" (10) and "Slightly influential" (10) covers 20, which is 2/3 of the participants. The result was quite promising for us in terms of Attelia's power efficiency.

Table 10: Summary of The Post-Experiment Survey (1)

Question	“Strongly disagree” (1)	“Disagree” (2)	“Neutral” (3)	“Agree” (4)	“Strongly agree” (5)	Average	Std. Dev.
1	6	10	5	7	2	2.6	1.2
2	3	4	7	15	1	3.2	1.1
3	2	3	7	15	3	3.5	1.0

- *Question (1): Did you realize any differences between several different notification strategies that changed daily, excluding “No notification” strategy?*
- *Question (2): Do you think the differences in Q(1) affected your mental workload?*
- *Question (3): Do you think the differences in Q(1) affected your response time to answer the quiz?*

Table 11: Summary of The Post-Experiment Survey (2)

Question	“Not at all influential” (1)	“Slightly influential” (2)	“Somewhat influential” (3)	“Very influential” (4)	“Extremely influential” (5)	Average	Std. Dev.
4	10	10	6	4	0	2.1	1.0
5	14	9	6	0	1	1.8	1.0

- *Question (4): With our software, did you see any difference in your phone’s battery life?*
- *Question (5): With our software, did you observe any performance degradation in your phone?*

8. Discussion

Our two types of user studies validated the effectiveness of Attelia, both in a controlled “in-lab” environment and in users’ in-the-wild real environments. With Attelia’s real-time breakpoint detection, notifications at the detected breakpoints resulted in 33% less cognitive load than notifications shown at random timings. The breakpoint-timing notification also resulted in 33% less frustration score than notification at random timings. Further, the response time for the notification was reduced by 12% in the breakpoint-timing notification. After getting these promising results, now we discuss several future research opportunities.

Firstly, further investigation on “insensitive” users (“WWL-insensitive user” and “FRU-insensitive users”) are the first interesting research topic for us. The opportunity includes possible real-time detection of which cluster the user belongs to, and investigation on possible other notification adaptation scheme for them so that their cognitive load can be lowered.

Second research opportunity is further system improvement with the model personalization technique. In this paper, we used a single model (decision tree) which are commonly used for the all participants. For example, active learning and a longitudinal user study are the possible next steps.

Deployment of the Attelia service to the “real notification system” on the smart phone operating system, including Android OS, is yet another challenging task for us. Our user study with the current implementation used our own artificial interruptive notifications due to the access limitations to the “real notifications” of the “real applications” inside Android OS. Meanwhile, Attelia is ready for exporting its “interruptibility API” based on the results of real-time breakpoint detection. Other applications can utilize Attelia’s API, through standard Android IPC mechanism.

In the recent mobile, ubiquitous, and wearable computing, users own and carry multiple mobile and wearable devices, including smart watches, smart bands, and smart glasses in addition to smart phones. The Attelia prototype already works on Google Glass and other versatile Android-based devices, including Android Wear watches, Android-based notebooks, and Android-based digital cameras. We hypothesize that, as long as such devices have any kind of user interface with which user interacts with the devices, our “application usage as a sensor” approach is applicable to those devices. Furthermore, expanding our research to the period when the user is not actively manipulating the devices is a very attractive research area. By using several other types of sensors on the devices, such as GPS, WiFi, Bluetooth, accelerometer, and proximity sensor, our breakpoint detection can be extended into such periods of users’ lives with those devices.

9. Related Work

In early work on finding appropriate moments for interruption, Horvitz et al. inferred interruptibility accurately in desktop computing environments, by using context information, such as interaction with computing devices, visual and acoustical analyses, and online calendars [25]. For this, recognition was performed in a posteriori manner.

Work by Begole et al. [26], Horvitz et al. [27] are in the first generation of systems with real-time model construction and detection of interruptibility although their systems used dedicated custom hardware. Iqbal et al.[28] built OASIS which defers desktop-based notifications until suitable timings of interruption were detected in real-time. They focused on the detection of breakpoints [8], based on user interactions with an application and provided user annotations.

More recently, interruptibility research has focused on mobile devices. Ho et al. used wireless on-body accelerometers to trigger interruptions in the timing of user's switch between activities [29]. The authors found that the users' annoyance was minimal when interruptions were triggered at the moments of switching between activities. While their approach needs an external on-body sensor, Attelia uses only the smartphone. Fischer et al. focused on the interruptibility immediately after phone activities including completion of phone calls and text messages [30]. They found that the users tend to be more responsive to notifications after mobile phone activities than at random other timings. While the authors' approach focused on phone-related activities, our approach uses applications available on the market and installed on the phone, including phone-call and text messaging. Smith et al. focused on disruptive phone calls and took the approach of "mitigation" by automatically setting phone call ring tones to different modes, such as silent answering, declining, and ignoring [31]. A user study showed that their approach was useful, even with user concept drift. Their mitigation approach is orthogonal to our scheduling approach, thus a combination of both approaches is possible.

Hofte et al. used smartphones for interruptibility study. They used the experience sampling methodology on location, transit status, company and activities to build a model for interruptibility [32]. Also Pejovic et al. explored whether, and how, suitable moments for interruption can be identified and used in smartphones [33]. Based on "broader context" including activity, location, time of day, emotions and engagement, their InterruptMe system decides interruptibility. Their approach determines timing based on smartphone sensor data. In contrast, our approach relies on user interaction, focusing on the period while the user is actively manipulating the device. According to our power consumption measurement in Section 5, we found that the power overhead for our approach was significantly lower than the physical sensor approaches. Also as users will continue to have an increasing number of devices, we believe our approach will be more effective because our approach can be easily deployed to devices with and without physical sensors. Also their implementation relies on information on user's activity, such as work mode, emotion, and company, manually provided by the user in order to infer interruptibility. On the other hand, our system does not need any manually-provided information, simply relying on the UI events coming from the Android system.

10. Conclusion

In this paper, we proposed a novel middleware, Attelia, which detects opportune moments for interruptive notifications delivery in order to minimize the impact on user's cognitive load, towards the realization of attention-aware adaptation that maintains user's productivity. Attelia identifies such timings in real-time, only with the middleware on the mobile devices that users naturally use, without any external dedicate psycho-physiological sensors, and without any modifications into versatile applications installed in the mobile devices. Our two types of user study validated the effectiveness of Attelia. Throughout both in-lab user study and in-the-wild user study, the results showed that the interruptive notifications displayed at the timing of breakpoints detected by Attelia resulted in significant reduction in user's cognitive load, frustration score, and response time to the pop-up, compared to the notifications in the random timings.

Acknowledgements

This work was supported (in part) by Ministry of Education, Culture, Sports, Science and Technology (MEXT) Grant-in-Aid for the "Research and Development for Big Data Use and Application" and for the "Program for Leading Graduate Schools".

References

[1] Deloitte Touche Tohmatsu Limited, Deloitte global mobile consumer survey 2012 (Sep. 2012).
URL `http://www.deloitte.com/view/en_GB/uk/industries/tmt/telecommunications/global-mobile-consumer-survey-2012/`

[2] Google Inc., The new multi-screen world — think with google (Aug. 2012).
URL `http://www.google.com/think/research-studies/the-new-multi-screen-world-study.html`

[3] D. Garlan, D. Siewiorek, A. Smailagic, P. Steenkiste, Project aura: toward distraction-free pervasive computing, Pervasive Computing, IEEE 1 (2) (2002) 22 –31. `doi:10.1109/MPRV.2002.1012334`.

[4] P. D. Adamczyk, B. P. Bailey, 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, 2004, pp. 271–278.

[5] B. P. Bailey, J. A. Konstan, On the need for attention-aware systems: Measuring effects of interruption on task performance, error rate, and affective state, Computers in Human Behavior 22 (4) (2006) 685 – 708.

[6] M. Czerwinski, E. Cutrell, E. Horvitz, Instant messaging: Effects of relevance and timing, in: People and computers XIV: Proceedings of HCI, Vol. 2, British Computer Society, 2000, pp. 71–76.

[7] J. G. Kreifeldt, M. E. McCarthy, Interruption as a test of the user-computer interface, in: JPL Proceeding of the 17 th Annual Conference on Manual Control, 1981, pp. 655–667.

[8] D. Newtson, G. Engquist, The perceptual organization of ongoing behavior, Journal of Experimental Social Psychology 12 (5) (1976) 436–450.

[9] S. Gould, D. Brumby, A. Cox, V. González, D. Salvucci, N. Taatgen, Multitasking and interruptions: a sig on bridging the gap between research on the micro and macro worlds, in: CHI'12 Extended Abstracts on Human Factors in Computing Systems, 2012, pp. 1189–1192.

[10] F. R. Zijlstra, R. A. Roe, A. B. Leonora, I. Krediet, Temporal factors in mental work: Effects of interrupted activities, Journal of Occupational and Organizational Psychology 72 (2) (1999) 163–185.

[11] C. Speier, J. S. Valacich, I. Vessey, The influence of task interruption on individual decision making: An information overload perspective, Decision Sciences 30 (2) (1999) 337–360.

[12] S. T. Iqbal, E. Horvitz, Notifications and awareness: A field study of alert usage and preferences, in: Proceedings of the 2010 ACM Conference on Computer Supported Cooperative Work, CSCW '10, 2010, pp. 27–30. doi:10.1145/1718918.1718926.

[13] S. T. Iqbal, B. P. Bailey, 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, 2007, pp. 697–706. doi:10.1145/1240624.1240732.

[14] S. T. Iqbal, B. P. Bailey, Investigating the effectiveness of mental workload as a predictor of opportune moments for interruption, in: CHI '05 Extended

Abstracts on Human Factors in Computing Systems, CHI EA ’05, 2005, pp. 1489–1492. doi:10.1145/1056808.1056948.

[15] S. T. Iqbal, B. P. Bailey, Leveraging characteristics of task structure to predict the cost of interruption, in: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’06, 2006, pp. 741–750. doi:10.1145/1124772.1124882.

[16] J. Burke, D. Estrin, M. Hansen, A. Parker, N. Ramanathan, S. Reddy, M. B. Srivastava, Participatory sensing, in: In: Workshop on World-Sensor-Web (WSW 2006): Mobile Device Centric Sensor Networks and Applications, 2006, pp. 117–134.

[17] M. Hoshiba, O. Kamigaichi, M. Saito, S. Tsukada, N. Hamada, Earthquake early warning starts nationwide in japan, Eos, Transactions American Geophysical Union 89 (8) (2008) 73–74.

[18] E. Haapalainen, S. Kim, J. F. Forlizzi, A. K. Dey, Psycho-physiological measures for assessing cognitive load, in: Proceedings of the 12th ACM international conference on Ubiquitous computing, 2010, pp. 301–310.

[19] Machine Learning Group at the University of Waikato, Weka 3: Data Mining Software in Java.

[20] Samsung Electronics Co., Ltd., Galaxy Nexus.

[21] Monsoon Solutions Inc., Monsoon Power Moniter.

[22] B. Bailey, J. Konstan, J. Carlis, Measuring the effects of interruptions on task performance in the user interface, in: Systems, Man, and Cybernetics, 2000 IEEE International Conference on, Vol. 2, 2000, pp. 757–762 vol.2. doi:10.1109/ICSMC.2000.885940.

[23] B. P. Bailey, J. A. Konstan, J. V. Carlis, The effects of interruptions on task performance, annoyance, and anxiety in the user interface, in: IFIP

TC.13 Conference on Human Computer Interaction (Interact 2001)., 2001, pp. 593–601.

[24] S. G. Hart, L. E. Staveland, Development of nasa-tlx (task load index): Results of empirical and theoretical research, in: P. A. Hancock, N. Meshkati (Eds.), Human Mental Workload, Vol. 52 of Advances in Psychology, North-Holland, 1988, pp. 139 – 183. doi:http://dx.doi.org/10.1016/S0166-4115(08)62386-9.

[25] E. Horvitz, J. Apacible, Learning and reasoning about interruption, in: Proceedings of the 5th International Conference on Multimodal Interfaces, ICMI '03, 2003, pp. 20–27. doi:10.1145/958432.958440.

[26] J. B. Begole, N. E. Matsakis, J. C. Tang, Lilsys: Sensing unavailability, in: Proceedings of the 2004 ACM Conference on Computer Supported Cooperative Work, CSCW '04, 2004, pp. 511–514. doi:10.1145/1031607.1031691.

[27] E. Horvitz, P. Koch, J. Apacible, 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, 2004, pp. 507–510. doi:10.1145/1031607.1031690.

[28] S. T. Iqbal, B. P. Bailey, Oasis: A framework for linking notification delivery to the perceptual structure of goal-directed tasks, ACM Trans. Comput.-Hum. Interact. 17 (4) (2010) 15:1–15:28. doi:10.1145/1879831.1879833.

[29] J. Ho, S. S. Intille, 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, 2005, pp. 909–918. doi:10.1145/1054972.1055100.

[30] J. E. Fischer, C. Greenhalgh, S. Benford, 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, 2011, pp. 181–190. doi:10.1145/2037373.2037402.

[31] J. Smith, N. Dulay, Ringlearn: Long-term mitigation of disruptive smartphone interruptions, in: Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on, 2014, pp. 27–35. doi:10.1109/PerComW.2014.6815160.

[32] G. H. H. ter Hofte, Xensible interruptions from your mobile phone, in: Proceedings of the 9th International Conference on Human Computer Interaction with Mobile Devices and Services, MobileHCI '07, 2007, pp. 178–181. doi:10.1145/1377999.1378003.

[33] V. Pejovic, M. Musolesi, InterruptMe : Designing Intelligent Prompting Mechanisms for Pervasive Applications, in: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '14, 2014, pp. 395–906.