InterruptMe: Designing Intelligent Prompting Mechanisms for Pervasive Applications

Veljko Pejovic
School of Computer Science
University of Birmingham, United Kingdom
v.pejovic@cs.bham.ac.uk

Mirco Musolesi
School of Computer Science
University of Birmingham, United Kingdom
m.musolesi@cs.bham.ac.uk

ABSTRACT

The mobile phone represents a unique platform for interactive applications that can harness the opportunity of an immediate contact with a user in order to increase the impact of the delivered information. However, this accessibility does not necessarily translate to reachability, as recipients might refuse an initiated contact or disfavor a message that comes in an inappropriate moment.

In this paper we seek to answer whether, and how, suitable moments for interruption can be identified and utilized in a mobile system. We gather and analyze a real-world smartphone data trace and show that users' broader context, including their activity, location, time of day, emotions and engagement, determine different aspects of interruptibility. We then design and implement InterruptMe, an interruption management library for Android smartphones. An extensive experiment shows that, compared to a context-unaware approach, interruptions elicited through our library result in increased user satisfaction and shorter response times.

Author Keywords

Interruptibility; Mobile sensing; Context-aware computing; Machine learning.

ACM Classification Keywords

H.5.2. Information Interfaces and Presentation (e.g. HCI): User Interfaces; H.1.2. Models and Principles: User/Machine Systems

INTRODUCTION

Mobile phones are ubiquitous, highly personal devices: the number of mobile cellular subscriptions is almost equal to the number of people on the planet [1], and the phones are, for most of the day, with their owners as they go about their daily routine. Moreover, mobile phones are far from merely being voice communication devices. Modern smartphones enable always-on e-mail, online social network and instant messaging communication. Therefore, the mobile phone represents the most direct point of contact with a person.

UbiComp'14, September 13 - 17, 2014, Seattle, WA, USA

ACM 978-1-4503-2968-2/14/09...$15.00.
http://dx.doi.org/10.1145/2632048.2632062

While the attractive properties of mobile phones indicate that nowadays almost anyone can be reached at all times, the utility of the resulting interruptions to the user are debatable. Take instant messaging (IM), for example. While Garrett and Danziger argue that IM helps office workers to quickly obtain task-relevant information [14], Cutrell et al. find that irrelevant messages result in prolonged task completion times [9]. That interruptions at wrong times lead to reduced worker performance, increased errors and stress is by no means unique to IM; the effect has been observed with other means of communication as well [5]. However, while in the past people reorganized their work schedules, or sought isolated locations in search for uninterruptibility [30], pervasive technologies make it hard to escape unwanted interruptions. At the same time, nowadays a large gamut of smartphone applications tend to initiate contact with the user. From restaurant recommendation to health and behavior tracking, from personalized advertisements to video conferencing applications. These applications often run in parallel to each other, and to other services such as a battery monitoring service, alarm clock, SMS, and phone's voice service. The ever-increasing number of reasons to notify a user, together with no intelligence that would discern if the moment for interruption has come or not, threaten to create a cacophony of notifications that would be of little utility to the user.

In this work we tackle the problem of designing intelligent interruption mechanisms for mobile devices. We concentrate on sensor-enabled devices such as smartphones, since the sensors provide information about the surrounding context of the user, aspects of which include location, activity, collocation with other users, to name a few. We hypothesize that this context determines user's interruptibility, but at the same time we claim that the interruptibility is a complex property and should be discussed from various aspects. We collect and analyze real-world data sets of mobile user interruptibility, and through an extensive machine learning-based study identify links between the context and the presence of user's reaction, between the context and a timely reply, and between the context and the sentiment that a user has towards the interruption.

We then funnel our findings into a practical implementation of an interruption mechanism for mobile devices – InterruptMe, a library for intelligent user notification for the Android operating system. InterruptMe collects relevant sensor data for context recognition, builds classifiers for identifying opportune moments for interruption, and integrates with the Android notification mechanism. The classifiers are instantiated within the library, can be personalized for each user, and

can evolve over the course of the application lifetime. While designing the library we pay attention to restrictive factors in mobile computing, such as limited battery charge and computing resources. A month-long experiment with ten subjects shows that, compared to a context-unaware approach, InterruptMe notifications lead to a faster user response time, and more favorably ranked moments to interrupt.

RELATED WORK

The importance of timing an interruption has been recognized in psychology in the past: by analyzing task switching processes researchers has shown that interruptions coming in different phases of task execution lead to different levels of disruption [27]. Moreover, in their study of instant messaging, Cutrell et al. demonstrate that the main task users are working on can be significantly disrupted if IM interruptions occur at inappropriate times [9]. To gauge suitable moments for sending a message, Avrahami and Hudson devise a statistical model of user responsiveness to an IM interruption. Desktop PC interaction events, such as key press and mouse movement are used as features in a response time classifier [4].

Opportune moments for interruption can be more accurately determined if a user's wider context is taken into account. Thus, Horwitz et al. use an external camera and microphone to infer user's availability in an office setting [18]. The outcome of this work is a Bayesian network that infers the user-defined cost of interruption from audio-visual features. In another early work Fogarty et al. investigate which sensors should be constructed in order to efficiently infer interruptibility [12]. They embrace the *Wizard of Oz* technique, where human subjects simulate sensors by manually coding information from audio and video recordings. Despite the approach that allows examination of sensors that are yet to be implemented in reality, the authors find that simple sensors often provide sufficient information to infer interruptibility.

The above work dealt with a posteriori recognition of opportune moments to interrupt. First systems to implement on-the-fly construction of interruptibility models include Lilsys [6] and BusyBody [19]. However, both systems represent purely research platforms with custom hardware and the lack of support for overlying applications. Nevertheless, BusyBody proved valuable for refining the machine learning aspects of interruptibility management. Using BusyBody, Kapoor and Horvitz demonstrate a decision-theoretic approach for probe scheduling, so that a user interruptibility model can be built with the least number of interruptions [23]. Iqbal and Bailey developed OASIS, a system that allows notifications to be deferred until suitable points for interruption are identified. Desktop PC interaction features, such as switching to a mail client, application downloaded, and similar, are used in the moment classification [21]. However, unlike previous work that strived to establish a direct link between recorded or sensed features and interruptibility, Iqbal and Bailey concentrate on recognizing breakpoints in the tasks, as previous research identified task boundaries as the most suitable moments for interruption [3, 20].

Ho and Intille [17] investigate the interruption burden in case of mobile notifications. Their study uses on-body accelerometers, and triggers interruptions only when a user switches her activity. The authors find that moments of changing activity, as inferred by the accelerometers, represent times at which an interruption results in minimal annoyance to the recipient. In [37] Ter Hofte studies interruptibility with smartphones, yet does not employ mobile sensing, but builds a model of interruptibility from self-reported location, company and activity information. Fischer et al. demonstrate that interruptions coming immediately after the episodes of mobile phone activity, such as a phone call completion or a text message sending event, result in a more responsive user behavior [10]. Pielot et al. collected a data set of text messages exchanged via smartphones together with the associated phone usage context [31]. Time since the screen was on, time since the last notification, and similar features were used in a classifier that infers if the users is going to attend the message within a short time frame.

The existing work established user–phone interaction episodes as representable indicators of opportune moments to interrupt. These episodes represent natural task boundaries, e.g. a completed phone call, and consequently good moments to interrupt. In this work we do not restrict to identifying task switching moments, but hypothesize that many other opportunities can be uncovered if full mobile sensing is employed. Finally, the existing work concentrates on a post-mortem analysis of interruption traces, while we build a system that allows real-time mobile notification management on the smartphone.

REASONING ABOUT INTERRUPTIONS

Interruptions are an indispensable part of everyday life. Tasks that we perform consist of both individual and interactive activities, often without a strict boundary between the two. Interruptions are valuable for bringing in relevant information for the current task and for notifying the recipients of important changes [30]. The mobile phone represents a great platform for interactive applications, which can harness the opportunity of an immediate contact with a user in order to increase the impact of the delivered information.

Our goal is to develop mechanisms to identify *opportune moments* for mobile device-based interruptions. In opportune moments we maximize the success of delivering content. First, however, we must define what a successful interruption is, and form verifiable hypotheses about what constitutes the stimulus – reaction relationship. We consider a system where a stimulus comes in a form of a notification on a user's mobile device. A reaction to the notification happens when a user responds to the content delivered through the notification, and this reaction need not be instantaneous. Although broad, this definition does not embrace all mobile interruption cases, such as phone calls. However, it closely resembles email, SMS and online social network (OSN) interruptions.

Identifying Opportune Moments for Interruption

The success of delivering information through an interruption can be measured from various perspectives. First, we can investigate *responsiveness* and term an interruption successful if it induces a reaction from the subject. This measure is appropriate for cases where a piece of information should not

be overlooked or ignored, irrespective of the user's sentiment towards the timing of the interruption. For example, the goal of important public safety queries is for them to be reacted upon, whether or not the user preferred to be interrupted at the moment of notification or not. Yet, we can specify a notification goal further, so that a mere existence of a reaction is not sufficient, and the reaction has to arrive within a certain time interval. This flavor of responsiveness is relevant in the case of an interruption that reminds a user to take a medication, as its effectiveness falls off with the time delay between the prescribed time and the moment when the user reacts to the reminder. A significant body of previous work used responsiveness as the evaluation metric for interruption [4, 21, 10, 33]. Yet, responsiveness does not reveal whether the subject welcomed the interruption or not. The *sentiment* that a recipient has towards the interruption is especially important in marketing applications, where an obtrusive interruption can lead to unfavorable perception of the advertised brand [32]. Self-reported sentiment towards the moment of interruption was used in previous work by Fogarty et al. [13, 12] as well as by Ho and Intille [17] and Fischer et al. [11]. Since multiple definitions of interruption success figure in real world applications, we group the above metrics and consider the following objectives in our interruptibility model:

- **Reaction presence.** We aim to predict if a recipient will react to an interruption.
- **Time to reaction.** We aim to predict if a recipient will react to an interruption within a given time interval.
- **Sentiment.** We aim to predict a recipient's attitude towards a moment in which a notification comes.

A separate line of research considers the performance on the main task after an interrupted user reverts to it, as a metric for the timeliness of an interruption [15, 21, 34]. Our experiments were done "in the wild", and in general we cannot measure aspects pertaining to an unknown main task.

Context – Response Hypothesis

A mobile user's context is described in part by physical properties of the setting, such as the location, movement, and time. But for interruptibility analysis the context also includes the internal state of the user, such as his engagement with a specific activity, social environment and emotions. While previous work in the field of interruptibility modeling identified ties between the context in which a user is, and the related level of interruptibility [18, 19, 13, 12, 37], different views on what constitutes the relevant context were not considered. In this work, we capture following flavors of the context and hypothesize how they relate to notification reaction:

- **Notification context.** The context in which a notification is sent determines an interruption success.
- **Response context.** The context in which a reaction is recorded determines an interruption success.
- **Notification-response context change.** In case of a successful interruption, the context changes from the notification to the reaction.
- **Notification context variation.** The variation of context at the notification time indicates an interruption success.

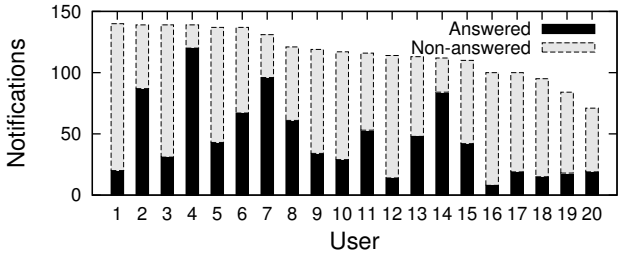


Figure 1. Per-user distribution of answered and unanswered notifications in the SampleMe dataset.

The existing literature largely considered the notification context, while Ho and Intille analyzed the notification context variation [17]. It is worth noting that these works did not involve smartphone sensing. In fact, to the best of our knowledge, our work is the first to investigate notification interruptibility in relation to smartphone-sensed context. It is also the first to explore the remaining two aspects of the context.

MOBILE INTERRUPTIONS DATA

To answer the research questions posed in the previous sections we rely on a real world data on human interruptibility. We acquire such data from SampleMe – an Android experience sampling method (ESM) application we designed to initiate interruptions, record the context, and collect user responses to the interruptions.

SampleMe Dataset

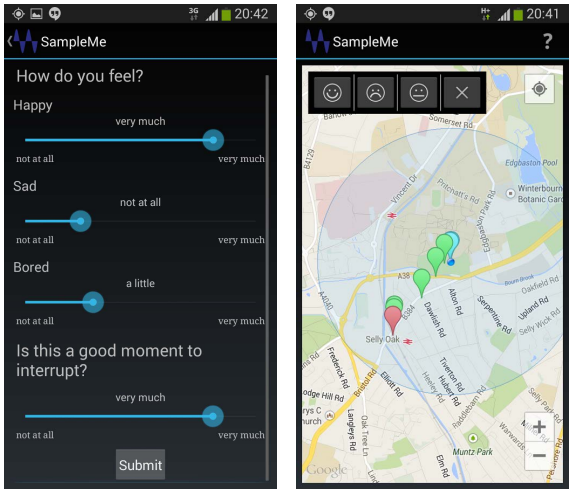
SampleMe is built on top of EmotionSense [25] and the SensorManager, TriggerManager and ESDataManager libraries [26]. The application notifies the user about a survey, which consists of questions related to user's attitude towards being interrupted, his location, activities, company, and emotions (Figure 2(a) shows the first page of the survey). The notifications are announced with a default ringtone/vibration from the mobile phone, and an icon shown in the notification bar. The application is also sensing user's GPS coordinates, Bluetooth and WiFi environment, and the phone's accelerometer. Context sensing is performed once a notification is sent to a phone, and once a user replies to the survey. In Table 1 we list the groups of features we record. Due to the limitations of Android sensing, for example, the inability to sense from a busy sensor, or an occasional failure to lock to the GPS, context data is not complete for all the notification instances.

Data collection was carried out for two weeks among 20 adult subjects, who received a modest monetary reward for their participation, where "participation" was defined as having the application running on their phones, and no requirements were made on actually reacting to notifications. SampleMe was running alongside other applications that the participants use on their phones. We recruited subjects from three continents, from both sexes, and from the age span 20 to 37 years old. Eight notifications were sent to each participant per day at random times between 8 am and 10 pm local time. All of the subjects persisted through the whole duration of the study.

To create an incentive to respond, but avoid forcing responses, we integrate an emotion map with the application. The map, such as the one shown in Figure 2(b), displays the geographic

Group	Features
Time	Time of day, weekend indicator, time into experiment.
Accelerometer	Mean, variance and mean crossing rate of the accelerometer readings. Activity variation around the notification time.[#] Activity change from notification to response.[*+]
Location	Descriptive location: "Residential", "Work", or "Public". Location change from notification to response.[*+] Bluetooth fingerprint change from notification to response.[*+] WiFi fingerprint change from notification to response.[*+]
Company[*]	Company indicator: "alone", "not alone".
Activity Engagement[*]	Descriptive activity: "Work related", "Leisure", or "Maintenance". How important is the activity? How interesting is the activity? How challenging is the activity? How skilled is the user wrt the activity? How concentrated the user is? User's desire to do something else.
Emotions[*]	How happy, sad, angry, frightened and neutral user is?

Table 1. Feature groups from the SampleMe dataset. Marked with * are features available for answered notifications only; marked with + are characteristic for a context change from notification to reaction; marked with # characterize variation of context at the notification time.



(a) Survey. (b) Emotion map.

Figure 2. SampleMe application screenshots.

distribution of emotion intensity. The density of displayed information increases with the number of surveys answered. The behavior with respect to interruptions varied significantly among users. In total 2334 notifications with surveys were sent, out of which 906 were answered. SampleMe was designed to automatically restart on a phone boot event, yet, if the battery was drained, or if a user forcibly quit the application, notifications would not be received. In Figure 1 we show the number of notifications received and reacted upon (surveys answered) for each of the 20 subjects. Each answered survey revealed the user sentiment towards the interruption, as one of the questions in the survey explicitly asked the participants if the current interruption came at the right moment.

We sample GPS both at the time when an interruption is sent, and when the corresponding survey is replied to. Using the OpenStreetMap API [28] we get addresses that correspond to the recorded coordinates in both cases. In the survey, one of the questions asks a user to label the current location as "Residential", "Work", or "Public". We assign that label to the address inferred from user's current coordinates. This is done for all answered surveys in the dataset. Next, we iterate over the set of addresses recorded at notification times, and in case that the same address is observed in the set of already labelled addresses from the survey answers, we back-propagate the label to the address recorded at a notification time.

MODELLING INTERRUPTIBILITY

To model the interruptibility of a mobile user we apply a series of machine learning techniques and test the assumptions about context and response we laid out in *Reasoning About Interruptions* Section. Our goal is to predict the actual outcome (interruptibility) AO_i at an interruption instance (notification that is sent out) i, using N sensed features $f_1, ..., f_N$ that describe the current context. We use a machine learning model $g((\vec{f}))$ that takes as an input a vector $\vec{f^i} = (f_1^i, ..., f_M^i), M \leq N$ of selected features' values at instance i to obtain the predicted outcome PO_i, i.e., $PO_i = g(\vec{f^i})$. We use the open-source WEKA [16] and MOA [7] machine learning toolkits to build the interruptibility models. Function g corresponds to an actual classifier, such as naive Bayesian, Bayesian network, boosting, or other.

Evaluation of Interruptibility Models

We use a trace-driven simulator to evaluate the performance of the models. The simulator input includes notification timing and content of related answers, and recorded contexts from the SampleMe dataset. At each instance i, the simulator queries a model of interruptibility $g(\vec{f^i})$ with the related sensed context, described by $\vec{f^i}$, to infer if the current moment is an opportune moment for interruption, i.e. to decide whether to activate the current notification or not. In case the predicted notification outcome (PO_i) is favorable, the notification is activated. If the actual outcome of the activated notification (AO_i) satisfies an objective function, which means that the user either reacted to the notification, reacted within a certain time frame, or reacted with a certain sentiment, we consider that the moment was indeed opportune, and that the interruption was successful. For each instance we inspect the activated notifications and keep track of all the instances where a notification is successful. At any point, we can calculate *precision*, i.e., the proportion of the instances recognized by the model as opportune that were indeed opportune, and *recall*, i.e., the proportion of all the opportune instances that our model labelled as opportune. We are predominantly interested in the precision as our main goal is to design a "calm" system that does not interrupt at inappropriate times, and we assume that opportunities for interruption are ample.

We evaluate different classifiers g, and depending on the hypothesis that is being tested, use different input features $\vec{f}$ and objective functions (as defined in *Reasoning About Interruptibility* Section). The first hypothesis we propose, i.e., the notification context determines the outcome, relies on time, accelerometer, and location groups of features from Table 1 (except for those features marked with [#], [+] or [*]). When testing the second hypothesis, i.e., whether the outcome of the

Classifier	Precision	Recall
AdaBoost	0.64	0.41
Bayesian net	0.58	0.37
Naive Bayes	0.57	0.52
Baseline	0.39	0.38

Table 2. Notification context determines reaction presence.

Classifier	Precision	Recall
AdaBoost	0.46	0.10
Bayesian net	0.54	0.04
Naive Bayes	0.52	0.04
Baseline	0.27	0.26

Table 3. Notification context determines reaction sentiment.

notification depends on the context recorded at the time of a user's response, we use all the features from Table 1 except for those marked with $^+$ or $^\#$. Features marked with $^+$ reflect the change in context between the moment of notification and the moment of response, and are used for testing our third hypothesis that the context changes from a notification to a reaction in case of a successful interruption. The final hypothesis we test is that the variation of context at the time of notification indicates the interruption outcome. In SampleMe we take fine-grained accelerometer readings five minutes before and up to five minutes after each notification. We calculate the activity variation level and use it as a feature in the analysis (feature marked with $^\#$ in Table 1).

Depending on one of the objective functions from the list presented earlier, we label the outcome AO_i of each of the instances in the dataset as a binary *successful* or *unsuccessful*. Thus, when considering reaction presence, we label as successful any instance where a user responded to the notification; when considering time to reaction, we label as successful any instance where a user responded no later than t_d after a notification; when considering reaction sentiment, we label as successful any instance where a user indicated that the interruption was not irritating beyond a given threshold. We first experiment with *batch learning*, where the dataset is treated as a bag of unordered notifications, a subset of which is used for classifier training. Next, we move to *online learning* where we preserve the order of notifications and let a classifier learn through exploration. Finally, we evaluate the practicality of the interruption modeling approach.

Batch Learning

We start by testing the hypothesis that the notification context determines the outcome of the interruption. For the outcome function, we first select reaction presence. From the WEKA toolkit we select naive Bayesian, Bayesian network, and AdaBoost classifiers [8]. We perform ten-fold cross validation, where 90% of the samples are used for classifier training; the remaining are ran through the simulator which infers the outcome of the notification based on the selected features ($\vec{f}$). Table 2 summarizes the precision and recall averaged over ten runs for both datasets. From the precision values we observe that of all the moments our model labelled as opportune for interruption, around 60% indeed resulted in a user reaction. The precision improves as the sophistication of the model is increased (from a naive Bayesian, over a Bayesian network, to a boosting-based classifier). To put this result in a perspective, we implement a baseline classifier that calculates the ratio of training set interruptions which resulted in a user reaction, and then in the simulator activates a notification with the probability that corresponds to that ratio. The baseline does not take the context into account, and performs significantly worse than any of the context-trained classifiers. We again test whether the notification context determines the reaction, this time with a different objective function – time to reaction. In Figure 3 we show the ability of the models to predict if a user will react within a given time threshold t_d. We observe that the laxer requirements with respect to the prediction horizon lead to a more precise prediction.

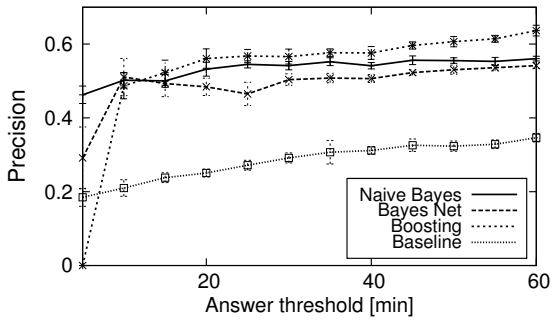


Figure 3. Notification context determines time to reaction. The x-axis denotes the time limit for the reaction prediction, i.e., whether the user reacts within the given time period.

Next, we experiment with inferring the sentiment that a user has towards interruption from the context recorded at the notification time. In SampleMe the sentiment was recorded as an answer to the question "Is this a good moment to interrupt?" on a four-point Likert scale with the following labels: *not at all*, *a little*, *somewhat*, and *very much*. For convenience, we define as opportune any moment that the user labelled as *a little* suitable, or above. In Table 3 we show the experimental results on the SampleMe dataset – the task is harder, and the precision is lower than in the case when only the presence of the reaction is predicted (Table 2), yet some improvement with respect to baseline is still observable.

We now evaluate the influence of the context at the time of response on the outcome of the interruption. In this case, we have a single meaningful outcome function – reaction sentiment. The other two outcome functions, the response presence and time to reply are trivially satisfied at the time when the context is recorded. We experiment with different definitions of an opportune moment to interrupt. First, we term a moment as opportune if in the corresponding survey the user labelled the moment as *a little* suitable, or above. Then, we tighten the definition by considering a moment to be opportune only if the user labelled it as *somewhat* or *very much* suitable. Finally, we define the moment as opportune only if the user labelled it as *very much* suitable for interruption. In Figure 4 we show the precision of inference with a changing threshold of acceptable interruption sentiment. The precision falls off as we tighten the definition of an opportune moment. This behavior is understandable, as the dataset gets heavily skewed towards moments that are not opportune. In Table 4 we show the distribution of recorded sentiment in the SampleMe dataset. In case the classifiers have to identify moments that are *very much* suitable for interruption, only 4.58% of the data points are labelled as successful. The small

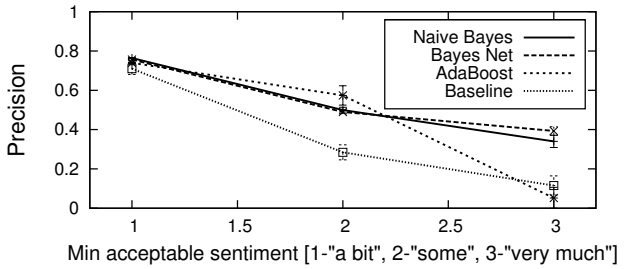


Figure 4. Response context determines reaction sentiment.

Sentiment	Not at all	A little	Some	Very much
Interruptions	1700 (72.84%)	377 (16.15%)	150 (6.43%)	107 (4.58%)

Table 4. Distribution of sentiment in the SampleMe dataset. A large majority of messages were not answered (we assign such interruptions as *not at all* appropriate) or arrived at moments that users labelled as *not at all* appropriate.

Feature	Opportune	Others	T-test
Acc change	M=0.45 SD=0.33	M=0.44 SD=0.34	t(720)=0.26, p=0.80
GPS change	M=448.65 SD=4690.76	M=150.63 SD=837.37	t(759)=0.95, p=0.34
BT change	M=0.33 SD=0.46	M=0.23 SD=0.42	t(341.27)=2.53, **p=0.01**
WiFi change	M=0.27 SD=0.38.76	M=0.22 SD=0.32	t(522.41)=1.87, p=0.06

Table 5. Notification-response context change is related to reaction.

Classifier	$t_d \to \infty$	$t_d = 180min$	$t_d = 120min$	$t_d = 60min$
AdaBoost	0.52, 0.36	0.44, 0.33	0.39, 0.29	0.05, 0.05
Baseline	0.39, 0.38	0.38, 0.38	0.37, 0.37	0.35, 0.35

Table 6. Notification context variability determines reaction presence. Each cell in the table contains the precision and recall values.

number of such points in the data set could also explain why boosting achieves low precision in this case, since for good performance boosting requires a richer training set.

We hypothesize that there is a relationship between notification-response context change and a user reaction. In SampleMe we record the context both at the time of notification, as well as at the time of a user response. We calculate the change in the reported GPS location, accelerometer features, Bluetooth and WiFi environment for the two context readings for each answered notification. The GPS change is expressed in meters, accelerometer change is represented by the intensity of a vector of accelerometer mean, variance and MCR between the two readings, and Bluetooth and WiFi change is expressed with the Jaccard distance between the sets of sensed devices recorded at the two context points. The only meaningful outcome function is sentiment towards interruption. We define a moment as opportune if the user labelled it as *a little* suitable for interruption or above. We split the dataset instances into two groups, opportune and others, according to the outcome, and perform a t-test between them in order to compare mean feature values in the two groups. A significant difference between the means would indicate that our hypothesis that the context changes from notification to reaction is correct. In Table 5 we show mean (M) and standard deviation (SD) of accelerometer, GPS, Bluetooth and WiFi change metrics for the two groups, along with the results of t-tests that compare the mean values of those metrics between the two groups[1]. A difference in the means is found only if the significance of the tests p is lower than a threshold (often equal to 0.05 in practice). In our case, the change in the Bluetooth environment is significantly different between opportune and non-opportune moments for interruption ($t(341.27) = 2.53, p = 0.01$). For any other feature, the difference between the means is not significant. The results should be taken with caution. The granularity of context change that we capture in our experiments might be insufficient to discern user's sentiment towards an interruption. On the other hand, establishing a bond between a more robust outcome, such as whether the reaction was present at all, would require frequent context sensing which was not performed in the SampleMe data collection experiment, due to energy constraints of smartphone sensing.

In the last hypothesis, motivated by Ho and Intille's earlier work in which the authors show that on-body accelerometers can be used to infer activity breakpoints, thus opportune moments for interruption, we hypothesize that the variation of context at the time of notification indicates opportune moments for interruption [17]. In order to measure context changes we take fine-grained accelerometer readings from five minutes before to five minutes after each notification. We then calculate a vector that consists of the mean, variance and the mean crossing rate in each one-minute window and measure the Euclidean distance between vectors recorded in subsequent time windows. We take the maximum observed distance as a measure of activity change at the notification time. We report the results with varying time-to-react threshold (t_d). We also test the reaction presence by setting t_d to infinity. In Table 6 we show the precision and recall of interruption inference. The only feature in the classifier is a real number that represents the measure of activity change. We show results for AdaBoost classifier, as other models performed significantly worse. Of the instances activated by AdaBoost 52% resulted in a reaction, which is significantly higher than the baseline. At the same time, AdaBoost captures, just like the baseline, about a third of all opportune interruption instances in the trace. However, unlike the previous work, we did not find a significant correlation between the interruptibility and activity change with a stricter definition of the opportune moment for interruption: once we require a reaction within a time frame of one, two or three hours, the prediction is not better than the baseline.

Online Learning

The classifiers we have considered so far are static: once trained, they are not modified with subsequent notification-outcome observations. Online classifiers allow model update

[1]We report statistics according to the American Psychological Association standards: T-test statistics are reported with the degree of freedom in parentheses, t-value, and the significance level.

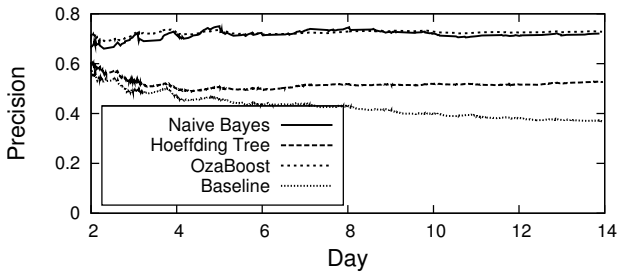


Figure 5. Notification context determines reaction presence (personalized online classification).

as new data instances are observed. This kind of learning is attractive for mobile interruption inference for the following reasons. First, human behavior and activities might change over time, for example as a person moves from one job to another. Second, in real-world applications training data are often unavailable. With online learning the classifier can evolve throughout the course of the experiment. Finally, and related to the previous two points, online classifiers are suitable for personalized interruptibility inferences. A classifier trained on data coming from a single person is likely to be more accurate in recognizing that person's behavior than a general classifier built upon data gathered from diverse populations [24]. The main drawback, however, is the limited amount of training data that we can collect from a single user. With online classifiers we can bootstrap a personalized classifier with a model built on a joint dataset, and then refine it with data coming from a single person as their data pours in.

We use online classifiers implemented by the MOA machine learning toolkit. The simulator sequentially goes over interruption data ordered in time; as in the previous experiments, at each data instance i the simulator queries a classifier g with the current context $\vec{f^i}$ to predict the outcome of the notification PO_i, thus whether the notification should be activated or not. Now, each activated notification, and its AO_i (which might be different from PO_i) are used for online classifier training. We investigate the performance of three online classifiers, naive Bayesian, Hoeffding tree, an online tree-based algorithm, and OzaBoost, an online version of a boosting approach [29]. The classifiers are trained to recognize the presence of the user reaction from the context recorded at the notification time. At the same time, we also test the potential of personalized classifiers by training a joint classifier for one day of the dataset, and then refining a separate copy of it on each of the individual user's data for the rest of the experiment. Figure 5 shows the precision of the classifiers (one day of common training and the first day during which the precision is not stabilized are not shown). All the tested classifiers perform better than the baseline, however, a simple naive Bayesian approach seems to be sufficient for online inference.

Interruptibility inference on a mobile device, such as a smartphone, is limited by the resources available on the platform, most notably battery capacity. Sensing context is often the most energy hungry operation in the interruptibility inference process. The battery life of continuously sensing smartphone reduces up to three times with accelerometer, and up to twenty times with the GPS sensor on [2]. Substantial energy savings can be achieved if reliable interruptibility inference can be done with low-power sensors. We use OzaBoost online classifier and investigate the precision and recall with different groups of context features used for classification. We restrict our analysis to time, location and activity features from Table 1. The results do not reveal a clear superiority of a single feature group or a combination of groups (Table 7). Relying on solely time features, which can be obtained with virtually no energy overhead, leads to a higher precision, compared to using only location or activity features. However, the portion of activated successful interruptions is lower than if only activity or location features are used.

	T	A	L	T & A	T & L	A & L	all
Precision	0.73	0.56	0.54	0.70	0.70	0.58	0.72
Recall	0.34	0.72	0.49	0.35	0.36	0.70	0.36

Table 7. Impact of different sensing modalities on accuracy of reaction presence inference from notification context. Different combinations of time (T), location (L) and activity (A) features are analyzed.

Practical Considerations

Our evaluation efforts so far have been geared towards the practical usability of an intelligent interruption mechanism. Thus, we evaluated the precision of delivering interruptions, rather than merely classifying is at successful or not. We also used online learning and personalized classification, and we evaluated the inference with different sensing modalities.

The results we obtained can be summarized as follows. First, the context at the notification time can be used to infer the reaction presence, and the user's sentiment towards reaction, with a significantly higher precision than the baseline (Tables 2 and3). In addition, unlike Ho and Intille did with body-worn sensors, with smartphone accelerometers we were not able to reliably detect activity boundaries, thus opportune moments for interruption. Second, the comparison of a batch-trained common classifier (Table 2), and personalized online classifiers (Figure 5) reveals higher precision of the latter. Moreover, a naive Bayesian classifier, which is straightforward to implement and naturally supports online classification, performs almost as well as more complex options. Finally, our analysis of classification with different sensing modalities (Table 7) reveals that the additional precision enabled by energy expensive sensors, such as GPS, is not necessarily justified from the resource efficiency point of view.

THE INTERRUPTME LIBRARY

Smartphones are naturally suited for self-contained interruptibility inference: sensing, learning and notification can all happen on the same device. Compared to inference performed on a remote server, locally executed inference algorithms can keep raw sensor data on the device, thus reduce data usage, energy consumption, and privacy concerns. Despite these benefits, the implementation of an interruption mechanism for smartphones is not straight-forward due to the inherent limitations of smartphones and peculiarities of mobile platforms. Obstacles include phone's limited energy storage and computing capabilities, bootstrapping, training and

dissemination of individual and group models of interruptibility, and operation in an environment that disfavors long-running tasks characteristic for server-side applications.

InterruptMe is our open-source Android library that enables identification of opportune moments for interruption[2]. The library allows the overlaying application to be notified of such moments, so that an appropriate action can be triggered. The Interruption Manager is the core of the library, holds a model of interruptibility and exposes a publish-subscribe API that notifies the application about opportune moments for interruption. To recognize these moments Interruption Manager maintains a model of interruptibility in the form of a classifier that establishes the relationship between the sensed context and interruptibility. InterruptMe periodically senses the context using a third party sensing library [25], and queries the classifier with the sensed data. The classifier is instantiated from a general purpose machine learning (ML) library we built[3], from which we use an online naive Bayesian classifier. If an opportune moment for interruption is recognized the Interruption Manager notifies the application via a callback. In addition, the manager allows the application to provide feedback about the outcome of the interruption. The feedback can include the user reaction or sentiment to the interruption, which will in turn be used to train and refine the interruptibility model, thus update the classifier. In the current implementation we train the classifier with the reported sentiment, more specifically, to recognize "very good" moments to interrupt.

InterruptMe is designed to be a light-weight library for intelligent notification management. InterruptMe library consists of 926 lines of Java code. The general purpose ML library it uses consists of 954 lines of code. The memory footprint of InterruptMe on an Android Nexus 4 phone accounts to 13.195 MB and around 48 thousand objects, compared to 14.070 MB and around 60 thousand objects allocated by a stub application running the WEKA Android port. The frequency of smartphone sensing and interruptibility inference in InterruptMe can be configured in order to save energy. In addition, InterruptMe harnesses the SensorManager library's ability to reconfigure sensing once the battery level is below a certain threshold. In our current implementation we use time and activity features from Table 1. Android OS disfavors long-running applications. Therefore, we instantiate the library in a background service. The service is periodically checked, and re-instantiated if needed. For persistence, InterruptMe keeps a copy of the interruptibility model in a JSON-formatted file on the phone. The same file can also be used to initialize a classifier for a user that just joined the system. The general goal, however, is to support the evolution of a common model for multiple users, and subsequent personalization of each of the model's copies.

EXPERIMENTAL EVALUATION

To evaluate the ability of InterruptMe to recognize opportune moments for interruption, we embed the library within the SampleMe application. The library communicates with a notification manager and decides on the notification delivery timing. InterruptMe evolves a personalized interruptibility classifier on the mobile, however, we also allow the library to override the intelligent interruption model and send notifications randomly, as in our initial SampleMe trial.

[2] `bitbucket.org/veljkop/intelligenttrigger`

[3] `github.com/vpejovic/MachineLearningToolkit`

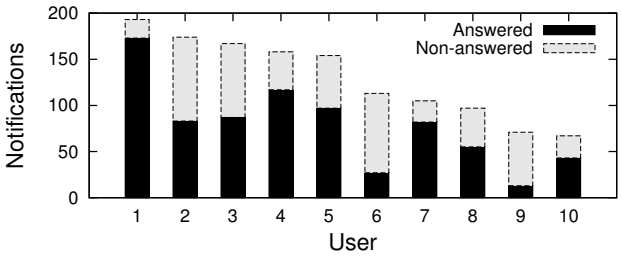


Figure 6. Per-user distribution of answered and unanswered notifications in the N of 1 trial.

We ran a month long trial of SampleMe with 10 subjects, eight male and two female, aged between 22 and 26, all graduate students. The application was augmented with the InterruptMe library, but the other aspects of the original SampleMe, including the emotion map screen and the form of notification, were preserved. To ensure that SampleMe surveys are answered honestly the students were given a course assignment that involved analyzing the data trace collected by their phones. There were no requirements on the amount of individual data needed for a successful assignment completion, and the subjects were told to use their phones as usual during the experiment. To compare the utility of InterruptMe managed notifications with those that were received at random moments we perform an *N of 1 randomized trial* [35]. In such an experiment, we alternate days when notifications about surveys are managed through the InterruptMe library with days when notifications are received randomly. In the original SampleMe run we noticed that individual reactions to notifications differ drastically (see Figure 1). The N of 1 trial helps us avoid the bias in response reactions that would skew the results if we were to employ a test method where the users are split into two groups according to the notification policy.

Each phone starts with a clean slate model of interruptibility that is trained every time a user reacts to a notification, i.e., fills out a survey. In this trial, our goal is to identify opportune moments for interruption as defined by a user's sentiment. We train the classifier with a positively labelled instance in case a user states that the moment is "very good" to interrupt; otherwise, we train the classifier with a negative label. In addition, non-answered notifications are labelled as non-opportune moments for interruption once a new notification comes in. We set a minimum interval of 10 minutes between any two successive notifications. We also limit the maximum number of notifications per day to 10, while activate random moment notifications with a probability that leads to six expected notifications per day. A total of 1285 notifications were received over all participants, out of which 763 resulted in a completed survey. The sample is approximately equally divided between random and InterruptMe-guided notifications. The distribution of notifications per user is shown in Figure 6. Due to technical problems, some users did not receive all the notifications in the first ten days of the experi-

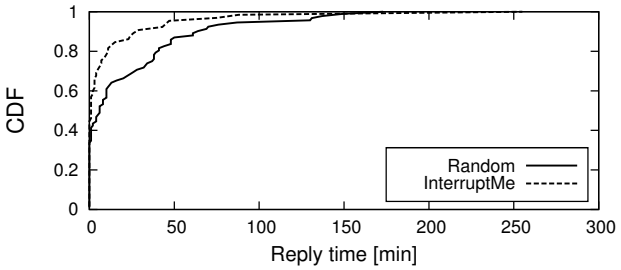
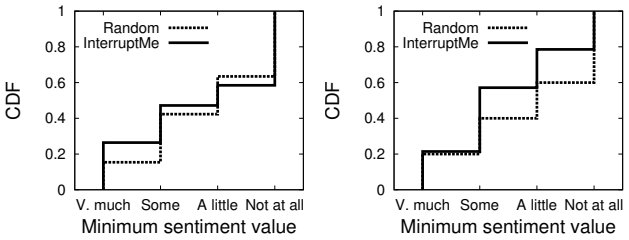


Figure 7. CDF of time to respond to a notification.



(a) All messages. (b) Isolated messages.

Figure 8. CDF of minimum reported sentiment.

ment, however, the losses were evenly distributed between the notification groups. In this analysis, we allow three weeks for the InterruptMe classifier to be trained, and perform the analysis on the data coming from the last week of the experiment.

Responsiveness

We first examine the time users took to react to a notification. In Figure 7 we show a CDF of reaction times, for answered surveys of all the users, for both methods of notification. InterruptMe-based notifications result in the response time of 12 minutes on the average, while random notifications on the average take 22 minutes to respond to. The significance of the difference was confirmed with a t-test: $t(141.02) = 1.90, p = .06$. The distribution also shows a non-negligible portion of surveys that are answered after a long delay. Such delayed answers can lead to a false impression when it comes to InterruptMe's ability to identify opportune moments for interruption. A survey answered in a moment long after the notification has been received will not provide information about the user's sentiment towards the original interruption moment. Moreover, we believe that the lack of a timely answer is an indicator that the original moment was not a good moment to interrupt. In the rest of the section, when considering a user-reported sentiment, we concentrate on surveys that were answered no later than ten minutes after the corresponding notification was received. Ten minute is also the minimum interval between consecutive interruptions in our experiment, thus by restricting the maximum response delay to this value we ensure that all notifications are treated as independent samples of interruptibility and are not directly affecting (overwriting) each other.

Sentiment towards interruption moments

The InterruptMe classifier is trained to recognize moments that users labelled as "very good" to interrupt at. When considering the correctness of the classifier we have to take into

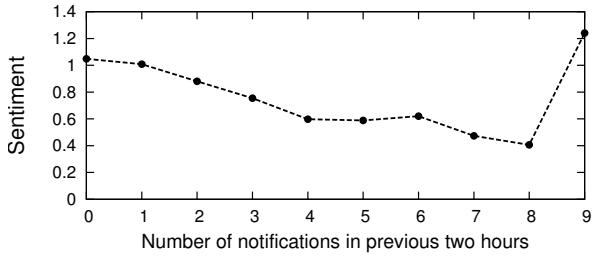


Figure 9. Mean sentiment towards an interruption versus number of interruptions in the preceding two-hour period for all the notifications sent during the experiment. The sentiment towards the last received notification falls off as the user copes with an increasing number of interruptions in a limited preceding time period. A notification can be preceded by at most nine other notifications in a day. A closer investigation reveals that when preceded by nine other notifications in a two-hour window the notification often lingers for a substantial time before being answered, therefore, acting like an isolated notification.

account that an answer to the question about current interruptibility is given on a four-point Likert scale. Thus, higher the user rated the moment, the better the classifier is. In the application we recorded the sentiment as an integer taking its value from zero, indicating a "not at all" good, to three, indicating a "very good" moment to interrupt. We are aware that the relationship among the labels need not be a linear one. However, the users were required to input the value via a slider where interruptibility labels were linearly ordered, thus, for the sake of further analysis we consider the reported interruptibility as a linear numeric variable. The mean value of reported interruptibility is higher, albeit not on the 90% level of significance, for InterruptMe-based notifications than for randomly scheduled ones: 1.32 versus 1.21, on a [0-3] scale. In Figure 8(a) we show a CDF of the minimum user reported sentiment towards the interruption moment. InterruptMe-based answered notifications are more favorably received, with 26.4% of them being marked as "very good" moment to interrupt, compared to 15.4% for randomly scheduled notifications. The difference diminishes as we relax the requirement for the minimum sentiment value, as seen by a higher portion of random messages received with the minimum sentiment value of "a little" interruptible or better.

Amount of Interruptibility

Interruptibility is often associated with user's frustration [13, 11]. In this experiment we did not explicitly ask users about their frustration and annoyance, but we suspect that these are reflected in the way users answer the moment sentiment question in the survey. We hypothesize that the recent exposure to interruptions determines user frustration, therefore, sentiment towards interruptions in the experiment.

In Figure 9 we plot the mean sentiment for an interruption moment versus the number of interruptions received in a two-hour time period that preceded the moment. In this general analysis we do not evaluate the behavior of the InterruptMe classifier, thus we use the whole month-long data set, and we also account for non-answered messages by assigning them a value zero - "not at all" suitable for interruption. The number of notifications preceding the current one is limited to nine,

as we have capped the total number of surveys per day to ten. The figure shows that isolated notifications tend to be more favorable than the ones received after a large number of recent notifications. For example, isolated notifications lead to more than twice as favorable sentiment, as compared to notifications that are preceded by eight other notifications in the two-hour interval. The only exceptions are notifications received after nine others. A closer investigation revealed that these are answered after a much longer delay (avg. answer time 71 minute) than the other notifications (avg. answer time 24 minutes), essentially acting like isolated notifications.

In contrast to random-based notifications that are received at random moments uniformly picked at the beginning of each day, in the InterruptMe-based real-time notification system a decision to trigger a notification is made dynamically and depends on the sensed context. However, the context can remain in the same, favorable, state for an extended period of time, in which case our application will deliver notifications back-to-back, and, as shown in Figure 9, exhaust the interruptibility of the user, which manifests through a lower reported sentiment. Note that the same "grouping" effect is unlikely to happen with context-oblivious random interruptions.

To evaluate the ability of InterruptMe to recognize opportune moments, but control for the effect of the amount of user interruptibility, we now restrict the dataset to notifications that were delivered in isolation, i.e., when no other notifications appeared for at least two hours before the current notification time. Figure 8(b) shows the CDF of user-reported interruptibility in this case. Compared to Figure 8(a), the gap between InterruptMe-based and random notifications is even larger, with the mean value of sentiment in the former case being 1.57, and 1.20 in the latter; however, the difference is still not on the 90% level of significance. In addition, the median sentiment towards InterruptMe-selected opportune moments is "somewhat" suitable for interruption, whereas it is "a little" suitable in the case of randomly selected moments.

DISCUSSION AND LIMITATIONS

Human interruptibility is a complex multifaceted issue, and in this study we investigate it by observing different outcomes of an interruption, and capturing a range of contextual features. Still, the complexity of the problem necessitates that certain views on the interruptibility were not considered in the study design. InterruptMe was designed to recognize if an opportune moment for interruption has come. While we indeed get a higher overall sentiment and a faster response to mobile notifications scheduled through InterruptMe, we also identify the frequency of interruption as a key parameter influencing a user sentiment towards interruption. To the best of our knowledge the existing work in the area of mobile HCI has not consider the amount of interruptibility, yet. The impact of the frequency of interruption has been investigated by Speier et al [36], where the authors show detrimental impact of frequent interruptions on the on the main task performance. While we have no means of measuring the main task performance in our non-controlled study, we hypothesize that the awareness of the performance degradation leads to an increased frustration and lower reported sentiment.

In this study we were predominantly motivated with the case where the content of the message cannot be adjusted to the user's interest, e.g. a smoking cessation digital behavioral change intervention (dBCI) must deliver a potentially annoying content. Therefore, we did not consider neither the origin nor the content of the interruption when identifying opportune moments. Often, however, humans judge their interruptibility based on the interrupter and the usefulness of the delivered information [30]. The content is also important, as its relevance to the current task impacts the demand it has on the main task the user is working on, as well as on how favorable the interruption will be [15, 22]. When it comes to mobile interruptions, Fischer et al. show that content's relevance, entertainment, actionability, and interest determine user's attitude towards interruptions [11]. In a large scale study of mobile notifications, Shirazi et al. find that notifications are considered important if they are about specific people or events [33]. Juxtaposing ours with the two studies by Fischer and Shirazi points out the importance of joint consideration of interruption design and delivery processes.

Finally, we design our study to measure *responsiveness* – a user's feedback reaction to a notification. Responsiveness is an important aspect of interruptibility, especially for communication applications [4]. *Attentiveness* describes the level of attention that the user has towards an incoming interruption. Detecting high attentiveness moments can be valuable for applications that benefit from the fact the user merely acknowledged the interruption, such as reminders. Attentiveness is less suited to be captured by a smartphone, since it does not necessarily elicit a recordable reaction. Nevertheless, a recent study by Pielot et al. established a link between some manifestations of attentiveness and features related to user's interaction with the phone [31].

CONCLUSIONS

In this paper we presented the design, implementation and evaluation of InterruptMe, a smartphone library that empowers an overlying application with personalized, evolving intelligent interruption models. The design of the library is based on an in-depth study of human interruptibility. We found that from the systems design point of view, a simple resource efficient online learner can serve as a basis for recognizing opportune moments for interruption. A month long experimental evaluation of InterruptMe demonstrated that our library represents a good starting point for identifying opportune moments for interruption. The experiments also revealed that such moments cannot be considered in isolation, and that users' sentiment towards an interruption depends on the recently experienced interruption load. In future, we plan to examine more sophisticated models that take interruption load into account, and consider the interruption content and originator when it comes to opportune moment identification.

ACKNOWLEDGMENTS

The authors would like to thank the participants of the SampleMe study. This work was supported through the EPSRC grant "UBhave: ubiquitous and social computing for positive behaviour change" (EP/I032673/1).

REFERENCES

1. *ITU World Telecommunication/ICT Indicators Database*, 2013.
2. F. B. Abdesslem, A. Phillips, and T. Henderson. Less is More: Energy-Efficient Mobile Sensing with SenseLess. In *MobiHeld'09*, Barcelona, Spain, August 2009.
3. P. D. Adamczyk and B. P. Bailey. If Not Now, When?: The Effects of Interruption at Different Moments Within Task Execution. In *CHI'04*, Vienna, Austria, April 2004.
4. D. Avrahami and S. E. Hudson. Responsiveness in Instant Messaging: Predictive Models Supporting Inter-Personal Communication. In *CHI'06*, Montreal, QC, Canada, April 2006.
5. B. P. Bailey and 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):685–708, 2006.
6. J. Begole, N. E. Matsakis, and J. C. Tang. Lilsys: Sensing Unavailability. In *CSCW'04*, Chicago, IL, USA, November 2004.
7. A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer. Moa: Massive Online Analysis. *The Journal of Machine Learning Research*, 11:1601–1604, 2011.
8. C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, NY, USA, 2006.
9. E. Cutrell, M. Czerwinski, and E. Horvitz. Notification, Disruption, and Memory: Effects of Messaging Interruptions on Memory and Performance. In *INTERACT'01*, Tokyo, Japan, 2001.
10. J. E. Fischer, C. Greenhalgh, and S. Benford. Investigating Episodes of Mobile Phone Activity as Indicators of Opportune Moments to Deliver Notifications. In *MobileHCI'11*, Stockholm, Sweden, August 2011.
11. J. E. Fischer, N. Yee, V. Bellotti, N. Good, S. Benford, and C. Greenhalgh. Effects of Content and Time of Delivery on Receptivity to Mobile Interruptions. In *MobileHCI'10*, Lisbon, Portugal, September 2010.
12. J. Fogarty, S. E. Hudson, C. G. Atkeson, D. Avrahami, J. Forlizzi, S. Kiesler, J. C. Lee, and J. Yang. Predicting Human Interruptibility with Sensors. *ACM Transactions on Computer-Human Interaction*, 12(1):119–146, March 2005.
13. J. Fogarty, S. E. Hudson, and J. Lai. Examining the Robustness of Sensor-Based Statistical Models of Human Interruptibility. In *CHI'04*, Vienna, Austria, April 2004.
14. R. K. Garrett and J. N. Danziger. IM= Interruption management? Instant messaging and disruption in the workplace. *Journal of Computer-Mediated Communication*, 13(1):23–42, 2007.
15. T. Gillie and D. Broadbent. What makes interruptions disruptive? a study of length, similarity, and complexity. *Psychological Research*, 50(4):243–250, 1989.
16. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.
17. J. Ho and S. S. Intille. Using Context-Aware Computing to Reduce the Perceived Burden of Interruptions from Mobile Devices. In *CHI'05*, Portland, OR, USA, April 2005.
18. E. Horvitz and J. Apacible. Learning and Reasoning about Interruption. In *ICMI'03*, Vancouver, Canada, November 2003.
19. E. Horvitz, P. Koch, and J. Apacible. BusyBody: Creating and Fielding Personalized Models of the Cost Interruption. In *CSCW'04*, Chicago, IL, USA, November 2004.
20. S. T. Iqbal and B. P. Bailey. Understanding and Developing Models for Detecting and Differentiating Breakpoints during Interactive Tasks. In *CHI'07*, San Jose, CA, USA, April 2007.
21. S. T. Iqbal and B. P. Bailey. Effects of Intelligent Notification Management on Users and Their Tasks. In *CHI'08*, Florence, Italy, April 2008.
22. S. Kalyanaraman, J. Ivory, and L. Maschmeyer. Interruptions and Online Information Processing: The Role of Interruption type, Interruption Content, and Interruption Frequency. In *Proc. of 2005 Annual Meeting of International Communication Association*, New York City, NY, USA, May 2005.
23. A. Kapoor and E. Horvitz. Experience Sampling for Building Predictive User Models: A Comparative Study. In *CHI'08*, Florence, Italy, April 2008.
24. N. D. Lane, Y. Xu, H. Lu, S. Hu, T. Choudhury, A. T. Campbell, and F. Zhao. Enabling Large-scale Human Activity Inference on Smartphones using Community Similarity Networks (CSN). In *UbiComp'11*, Beijing, China, September 2011.
25. N. Lathia, K. K. Rachuri, C. Mascolo, and P. J. Rentfrow. Contextual dissonance: Design bias in sensor-based experience sampling methods. In *UbiComp'13*, Zurich, Switzerland, September 2013.
26. N. Lathia, K. K. Rachuri, C. Mascolo, and G. Roussos. Open source smartphone libraries for computational social science. In *2nd Workshop on Mobile Systems for Computational Social Science (MCSS'13)*, Zurich, Switzerland, September 2013.
27. Y. Miyata and D. A. Norman. Psychological Issues in Support of Multiple Activities. *User Centered System Design: New Perspectives on Human-Computer Interaction*, pages 265–284, 1986.

28. OpenStreetMap API. `http://http://wiki.openstreetmap.org/wiki/API`, 2013. Accessed: 2013-05-23.

29. N. C. Oza. Online Bagging and Boosting. In *SMC'05*, Waikoloa, HI, USA, October 2005.

30. L. A. Perlow. The time famine: Toward a sociology of work time. *Administrative science quarterly*, 44(1):57–81, 1999.

31. M. Pielot, R. de Oliveira, H. Kwak, and N. Oliver. Didn't You See My Message? Predicting Attentiveness to Mobile Instant Messages. In *CHI'14*, Toronto, ON, Canada, April 2014.

32. R. Rettie, U. Grandcolas, and B. Deakins. Text Message Advertising: Response Rates and Branding Effects. *Journal of Targeting, Measurement and Analysis for Marketing*, 13(4):304–312, 2005.

33. A. S. Shirazi, N. Henze, T. Dingler, M. Pielot, D. Weber, and A. Schmidt. Large-Scale Assessment of Mobile Notifications. In *CHI'14*, Toronto, ON, Canada, April 2014.

34. T. Short, A. Rosenfeld, J. Goldbeck, and S. Kraus. CRISP - An Interruption Management Algorithm based on Collaborative Filtering. In *CHI'14*, Toronto, Canada, April 2014.

35. M. Sidman. *Tactics of Scientific Research: Evaluating Experimental Data in Psychology*. Basic Books, New York, NY, USA, 1960.

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

37. G. H. Ter Hofte. Xensible interruptions from your mobile phone. In *Mobile HCI'07*, Singapore, September 2007.