Designing Content-driven Intelligent Notification Mechanisms for Mobile Applications

Abhinav Mehrotra
University of Birmingham, UK
a.mehrotra@cs.bham.ac.uk

Mirco Musolesi
University of Birmingham, UK
University College London, UK
m.musolesi@ucl.ac.uk

Robert Hendley
University of Birmingham, UK
r.j.hendley@cs.bham.ac.uk

Veljko Pejovic
University of Ljubljana, Slovenia
veljko.pejovic@fri.uni-lj.si

ABSTRACT
An increasing number of notifications demanding the smartphone user's attention, often arrive at an inappropriate moment, or carry irrelevant content. In this paper we present a study of mobile user interruptibility with respect to notification content, its sender, and the context in which a notification is received. In a real-world study we collect around 70,000 instances of notifications from 35 users. We group notifications according to the applications that initiated them, and the social relationship between the sender and the receiver. Then, by considering both content and context information, such as the current activity of a user, we discuss the design of classifiers for learning the most opportune moment for the delivery of a notification carrying a specific type of information. Our results show that such classifiers lead to a more accurate prediction of users' interruptibility than an alternative approach based on user-defined rules of their own interruptibility.

Author Keywords
Mobile Sensing; Notifications, Interruptibility, Context-aware Computing.

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

INTRODUCTION
An increasing number of smartphone applications actively push information to the users. These services provide notifications about a variety of events, such as the arrival of an email, a new comment on a social network post, or a system update. Proactive services are indeed beneficial to the users, facilitating task switching and opening a number of information channels. However, due to the pervasive nature of smartphones, notifications often arrive at inconvenient moments.

UbiComp '15, September 07–11, 2015, Osaka, Japan.
© 2015 ACM. ISBN 978-1-4503-3574-4/15/09...$15.00
DOI: http://dx.doi.org/10.1145/2750858.2807544

Psychological studies have found that initiating interactions at inopportune moments can become a source of interruption [22, 32]. Such an interruption can adversely affect task completion time [8, 9, 23], error rate [5], and affective state of the user [3, 4]. Consequently, mobile notifications have a potential to drive smartphones away from Mark Weiser's vision of calm computing [30], which advocates "disappearing" technologies completely blended into our daily lives.

The interaction of a user with mobile notifications is indeed extremely complex and depends on numerous aspects. Fortunately, some of the aspects can be captured, and above all, *quantified*, by means of the embedded sensors that modern smartphones are equipped with. These sensors allow a mobile application to collect information about the user's day-to-day activities [7, 21], preferences [31], and the surrounding environment [19]. Past studies have used some of the sensed information to infer *opportune moments*, i.e., moments in which a user quickly and/or favorably reacts to a notification [10, 24]. More specifically, some important contextual factors that have been used to infer interruptibility include transitions [15], engagement with a mobile device [10], and, more generally, time of day, location and activity [24].

However, until now, the focus has been on the *context* in which a notification has been received and not on the actual *content* of the notification. After all, not all the notifications are disruptive [20]. It is *the relevance of the interruption content in the recipient's current context* that partly defines the disruptiveness of an interruption. For example, a chat notification from a friend can be extremely disruptive if delivered during a meeting. But, an email notification from a project collaborator might be acceptable to the user, and in some cases, considered very useful in the same context.

In this work we discuss how content and context can be used *together* in order to design intelligent non-disruptive notification mechanisms. More specifically, we investigate how users behave when they receive specific types of content through mobile notifications arriving at different times, and in different contexts. Unlike previous studies such as [24], we do not restrict ourselves to context information provided by mobile sensors only. Instead, to the best of our knowledge, for the first time we also take into account the type of information delivered and the social relationship between the information sender and receiver. It is worth noting that our work is predicated on the hypothesis that the acceptance of a mobile notification depends on *what*, e.g., the type, and the origin, of the information contained in a notification and *where*, the user's context in which the notification is delivered. We will consider a notification as accepted if it is handled (i.e., clicked to launch the respective application) by the user within a certain time of its arrival.

In order to test our hypotheses, we collect mobile notifications of 35 users for a time period of three weeks. We analyze these notifications to understand the key features that determine their acceptance. To the best of our knowledge, this is the first study to use notifications collected "in the wild" for such an analysis.

The outcome of our analysis can be summarized as follows:

- User's acceptance of a notification depends on the content and the originator of the notification and, not surprisingly, the time needed to respond to a mobile notification varies according to the user's physical activity level.
- Notification content, together with the sensed context, can serve as a basis for the design of machine learning classifiers that infer a user's response to a notification.
- Automated inference of opportune moments to interrupt, as with the above classifiers, outperforms subjective rules with which our subjects described their interruptibility.
- The inference of a user's interruptibility can be performed locally, in an online fashion, achieving a stable state (i.e., more than 60% precision) after nine days of training.

We conclude that inferring the right moment for interruption should be done in a holistic manner, jointly taking into account the context of the recipient, and the notification content. We show that a prediction model trained on a user's personal data performs far better than a generic prediction model trained on multiple users' data. Finally, we point out the importance of an automated approach, as humans remain inefficient at formalizing and communicating their preferences for mobile notifications in terms of predefined rules.

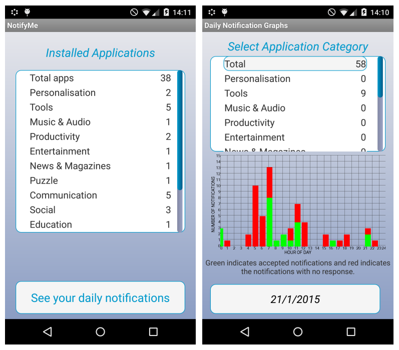
OPPORTUNE MOMENT TO DELIVER A NOTIFICATION

Mobile phones represent a great platform to increase the impact of information by delivering it in real time. At the same time, mobile notifications allow the users to be aware of newly available information. However, the wrongly timed notifications come with a cost of interruption to the on-going task [5]. This can cause negative emotions, reduce the value of the application from which such a notification was triggered. For example, the perception of a mobile marketing message, and, consequently, the whole brand that is being advertised could be impacted by the moment in which a message arrives [14, 27]. Thus, in order to provide the best value of information to the user, it becomes a necessity to deliver notifications at opportune moments.

What Defines an Opportune Moment?

In [6], Clark suggest that a user can respond to an interruption in four possible ways:

1. handle it immediately;
2. acknowledge it and agree to handle it later;


NotifyMe

Installed Applications

Total apps	38
Personalisation	2
Tools	5
Music & Audio	1
Productivity	2
Entertainment	1
News & Magazines	1
Puzzle	1
Communication	5
Social	3
Education	1

See your daily notifications

Daily Notification Graphs

Select Application Category

Total	58
Personalisation	0
Tools	9
Music & Audio	0
Productivity	0
Entertainment	0

Green indicates accepted notifications and red indicates the notifications with no response.

21/1/2015

Figure 1. NotifyMe application screenshots.

3. decline it (explicitly refusing to handle it);
4. withdraw it (implicitly refusing to handle it).

The first two categories of responses suggest that the interruption content is acceptable to the user. However, the second type of response indicates that *the interruption content is not relevant at the moment of its delivery* and, thus, it should have arrived after a certain time delay in order to be handled immediately. On the other hand, the responses with a decline and withdraw reflect that the interruption is not at all acceptable to the users.

We construct our hypothesis based on these possible responses of a user to an interruption. We hypothesize that a moment is opportune to deliver a notification only if the user handles the notification immediately. This does not mean that an intelligent interruptibility management system should just try to reduce the response time, it should also aim to increase the acceptance rate of notifications.

How to Predict an Opportune Moment?

Sahami et al. [28] demonstrated that the notifications triggered by applications from various categories are given different importance by users. At the same time, Pielot et al. [26] claimed that the user response is influenced by social pressure. This suggests that the user's decision about how to respond to a notification is made after looking at the notification title which gives a clue about the information contained in it. On the other hand, Grandhi et al. [12] suggested that by just relying on sensor based knowledge, interruption management systems often fail to infer whether the interruptions are disruptive or not because they do not take into account the sender of an interruption and the information that is being sent.

Thus, it is better to predict an opportune moment to deliver a notification by considering both notification content and the context in which it is delivered. To better understand the definition of opportune moments, let us consider a scenario in which a person at her workplace receives a notification regarding the arrival of an email from a colleague working on the same project and, separately, a new comment on one of her social networking posts. Given the circumstances, the user accepts the email notification leaving the social networking notification unattended. Later, on the way home, the user clicks on the social networking notification to read the new comment.

Feature	Description
Arrival time	Time at which a notification arrives in the notification bar.
Removal time	Time at which a notification is removed from the notification bar.
Response time	Difference between arrival and removal time.
Notification response	Whether the notification was clicked or not (boolean).
Sender application	Name and package of an application which triggers a notification.
Notification title	Title of a notification displayed in the notification bar.
Alert type	Signals used to alert the user for a notification: sound, vibrate, and LED.
Physical activity	Current activity of a user.
Location	Current location of a user.
Surrounding sound	Whether the user is in a silent environment or not (boolean).
WiFi connectivity	Whether the phone is connected to a WiFi or not (boolean).
Proximity	Whether the user was proximate to the phone in the last one minute or not (boolean).
Phone's status	Whether the phone was in use in the last one minute or not (boolean).
Ringer mode	Current ringer mode: sound, vibrate and LED.

Table 1. Description of features from the NotifyMe dataset.

In this scenario, since only one notification was read while the other was unattended, it is impossible to decide whether the sending moment is an opportune moment or not by using only the context of the user. Both notifications arrived at the same moment but were accepted in different contexts. It is the notification content that enables the recipient to decide whether to accept it or not in the current context. Thus, the notification content and the user's context together play an important role in identifying an opportune moment to deliver a notification. However, for privacy reasons it might not be feasible to exploit the content of notifications because they might contain extremely sensitive information[1]. Therefore, we propose using the notification title that contains more high-level and abstract, but at the same time, useful information about the category of information contained in the message itself. More specifically, the title can be used to classify the category of a notification. In addition, we capture a series of attributes about the user's physical context as well as the phone settings. These content and context data are then used to build a prediction model to predict the acceptance of a notification. The study was done in accordance with our institution's ethical research procedures, and all the participants were volunteers who provided their consent to collect data. The consent form itself for the data collection application was reviewed by the Ethics Board of our institution.

DATA COLLECTION

In order to study the influence of context and content on user response, we collected *in-the-wild* notifications from a set of users. More specifically, we developed an Android app called NotifyMe that runs in the background to unobtrusively monitor notifications and the context in which they are posted. It relies on Android's Notification Listener Service [1] to trace notifications, and uses Google's Activity Recognition API [2] and ESSensorManager [17] to obtain the context information. Table 1 lists the description of features captured by NotifyMe. It is worth noting that the *Alert Type* indicates the signals used by a notification to alert the user, whereas the *Ringer Mode* refers to the phone's ringer mode settings.

[1]We are aware that a large number of companies offering services such as social networking actually perform text processing from user-generated content. This type of analysis might not be possible in an academic context, and in any case, it raises significant privacy issues, especially for testing and validating the findings against ground-truth data, since the researchers have to access to the original text itself.

Question	Options
How would you rate the notification content?	Likert scale rating between 1 and 5 (1 = very annoying and 5 = very interesting).
Where would you like to receive notifications with similar content?	Home, workplace, other, anywhere and I don't want.
When would you like to receive notifications with similar content?	Morning, afternoon, evening, night, anytime and never.
How are you feeling?	Happy, sad, bored and annoyed.
Are you busy?	Yes and no.
Where are you?	Home, workplace, public, other.

Table 2. Questions and their options from NotifyMe questionnaire.

To infer the user response to a notification, NotifyMe checks whether the application that triggered the notification was launched after the removal time of that notification. Since most of the notifications are triggered by chat and email applications, users usually click on such notifications to read the full text and reply. Therefore, we assume that users click on the notifications that arrive at opportune moments. We are aware that our approach has limitations, because some notifications, which do not require further action, might not be clicked rather just seen and dismissed by the user.

Additionally, every day NotifyMe posts 12 notifications on the user's smartphone containing information that is randomly chosen from breaking news, weather update, and Facebook likes. These notification are triggered randomly every hour between 8.00 am and 8.00 pm. In this way we generate a consistent set of notifications for our analysis over all the users and we enhance the richness of the categories of information received by them. Furthermore, NotifyMe also collects subjective data from its users by triggering six questionnaires each day. A questionnaire comprises of six multiple-choice questions. Each questionnaire is triggered randomly in every two hours time window between 8.00 am and 8.00 pm. When the users are busy, NotifyMe allows them to decline the questionnaire notification by simply remove it from the notification bar; moreover, the application makes sure that it does not trigger another questionnaire for the next 30 minutes. The list of questions, along with the options included, are shown in Table 2.

Recruitment of the Participants

NotifyMe was published on Google Play Store and advertised at our University. It was installed by 35 participants without any monetary incentive. These participants come from both sexes, with the age span between 21 and 31 years. As shown in Figure 1, NotifyMe allows users to check the number of installed applications for each category and a bar graph of their notification acceptance rate for each hour of a day. We believe that displaying this information has a minimal interference on users' actual behavior in terms of notification acceptance, yet presents a valuable stimulus to make the users keep the app installed on their phones.

Ensuring Privacy Compliance

In order to allow the NotifyMe application to monitor notifications, the user has go give explicit permission as required by the Android operating system. Moreover, the application also shows a detailed user consent about the information that is collected. This ensures that the user is aware of the type of information captured by the application. Furthermore, we use notification content only to classify the category of information that a notification contains because such data can have severe privacy implications. We only store the notification's content category and discard the raw notification information.

DATASET

The data collection was carried out for a time period of 3 weeks from 35 users who installed the NotifyMe application. Overall, we collected more than 70,000 notifications and around 4,069 responses to questionnaires. Since, our objective is to predict the probability of the notification's acceptance by using a notification's content category and the user's context in which it is delivered, we classify the collected notifications according to the type of information they contain.

In order to classify a notification, we use the approach of mapping the notifications to the category of application from which they were triggered. The categories defined by the Google Play Store are too generic and no previous work has made any such list publicly available. Therefore, we manually categorize the applications from which the notifications were triggered.

Around 70% of the data set comprises of notifications triggered by chat and email applications. Since these two notification categories dominate over the combined notification counts of all the other application categories, we could not rely solely on the approach to classify notifications based on the category of applications with which they were triggered. Thus, we split chat and email notifications in the following four sub-categories based on the sender's relationship with the recipient of a notification:

1. Work: sender works or studies with the recipient;
2. Social: sender has a social tie with the recipient;
3. Family: sender is the recipient's family member or relative;
4. Other: sender not related to the recipient with the above relations.

In practical applications, this information might be extracted from social network platforms and/or inserted directly by the users (see also Discussion section). We believe that this methodology can be generalized: more fine-grained or different classification might also be possible.

We rely on the title of communication notifications in order to classify them in the above four classes. A communication notification's title comprises of the sender's name along

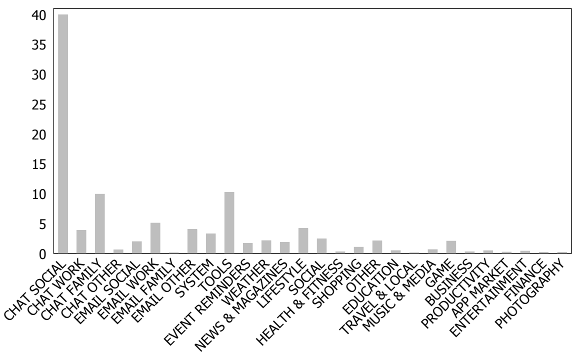


Figure 2. Percentage of notifications for each category and sub-category. The sub-categories are derived by using the recipient's relationship with the sender.

with a short description about the newly available information (such as "A new message from Alice", and "Alice sent you a new message"), or only the sender's name (such as "alice@gmail.com" or "Alice"). However, a few communication applications do not disclose any information about the sender in the notification title but only includes the description about the newly available information (such as "1 new message") or the application's name itself (such as "Telegram"). Therefore, we use the notifications that contain the sender's information in their title and discard the rest of the communication notifications. We then generate a list of unique titles[2] from the communication notifications of each participant and give these lists to their owners. We ask the participants to label each notification title with the following classes: work, social, family, and other. Participants were asked to provide multiple labels in case they have multiple relational links with the sender of a notification. For example, a colleague can also be a friend of a user, and an email from a job portal can be considered as both work and other.

Finally, we map the user-defined labels to the notification that owns the respective title. Since, there were a few notification titles that were labeled with multiple classes: 1) work and social, and 2) social and family, we choose a category based on the location in which the notification arrived. For example, if a notification that is labeled as both social and work, arrives at their workplace we consider it as a notification containing work related information. We define a category probability list for each location from our data and look for a label that has the highest value at the location in which the notification was triggered. The category probability list for each location is defined as follows:

1. Workplace: work, social, family;
2. Home: social, family, work;
3. Other: family, social, work.

Note that in order to infer these three location classes we sample GPS when a questionnaire is answered. We assign the label for the current location provided by the users as an answer to a question "Where are you?" in the questionnaire, to the sampled geo-coordinates. Since the location labels provided

[2]In order to avoid the users from labeling a title multiple times, we create this list containing notification titles without any duplication.

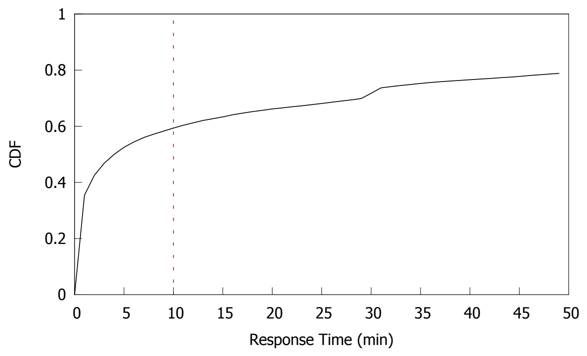


Figure 3. CDF of response time for notifications.

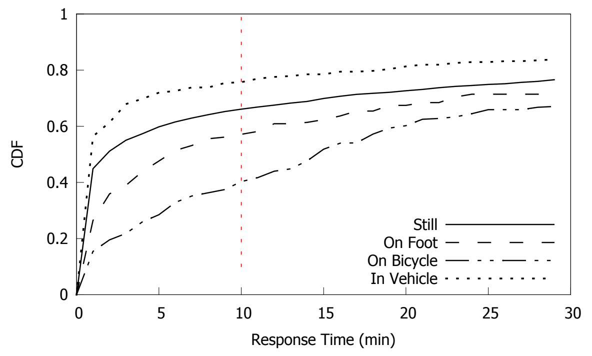


Figure 4. CDF of response time for notifications received while performing different activities.

by the users do not include any public location label, we exclude this location class. Additionally, we reverse geo-code the home address provided by the users at the time of registration and validate user-defined labels for these locations.

Out of 35 participants only 17 participants opted to label their communication notifications' title as discussed above. All of these participants continued to run NotifyMe application and actively responded to the questionnaires for at least 15 days. Therefore, in order to perform notification category-based analysis, we use around 25,000 notifications and 1,450 questionnaire responses of the participants who opted to label the communication notifications' title. Figure 2 shows the percentages of notifications that belong to each fine-grained category.

UNDERSTANDING INTERRUPTIBILITY

In this section we provide evidence that the content and context play an important role in influencing the response time and acceptance of a notification.

Time Delay in Response to a Notification

As discussed earlier, an intelligent interruptibility management (IM) system should define a moment as opportune to deliver a notification only if the user handles the notification in a given time interval. Therefore, it becomes necessary to define a threshold for time delay in response to the notifications by the users. The notifications that are responded to after this threshold will be classified as delivered in an inopportune moment. This approach is based on the strategy employed by the previous study [26] that predicts whether a user will view a message within the assumed threshold time.

We analyzed the complete dataset of notifications (containing around 70,000 notification samples) to draw a picture of the general response time for all notifications. Note that a notification is considered responded to when it is removed from the notification bar.

As shown in Figure 3, more that 60% of the notifications were clicked within 10 minutes from the time of arrival. The density of notifications increases with a high rate up to 10 minutes and after 10 minutes the rate starts stabilizing. This demonstrates that users handle most of the notifications by either clicking or dismissing them within a 10 minute period. Also, users do not interact with the notifications for a long time if they are not handled within 10 minutes. Overall, we can conclude that in general the maximum response time taken by a user to handle notifications that arrive at opportune moments is 10 minutes. However, it might vary according to the user's behavior. Therefore, in this study we choose 10 minutes as the *threshold time delay* for responding to a notification. At the same time, the goal of this analysis is not strictly dependent on the choice of this specific value. In other words, the aim is to provide a general design methodology for intelligent content-driven notification systems.

It is also worth noting that there is a sharp increase in the number of responses at around 30 minutes after the arrival. This is because some applications kill their notifications that do not get any response from the users up to a certain threshold. Our dataset shows that more than 70% of the notifications which were removed from the notification bar between 29 and 31 minutes were triggered by Google Now.

Impact of Context on Response Time

In the collected data we see that the notification response time can vary from a few seconds up to some hours. A notification delivered at an opportune moment might be responded to very quickly, but a notification can have a greater time delay if it arrives at an inopportune moment. We analyse the complete dataset of notifications to find the context modalities (i.e., the attributes of a user's context such as location, activity and others) that could indicate the response time for a notification.

We evaluate the response time of notifications with respect to three context modalities: location, activity, and surrounding sound. We find that the average response time of a notification does not vary with the user's location (i.e., home, workplace, and other) or the surrounding sound (i.e., silent or speaking). However, the data shows that the activity of users does impact the response time of notifications.

We rely on Google's Activity Recognition library [2] to classify a user's activity into four classes: 1. still – when the user is not moving; 2. on foot – when the user is either walking or running; 3. on bicycle – when the user is riding a bike; 4. in vehicle – when the user is travelling in a vehicle.

As shown in Figure 4, users are very quick to respond to notifications while they are traveling in a vehicle. Around 80%

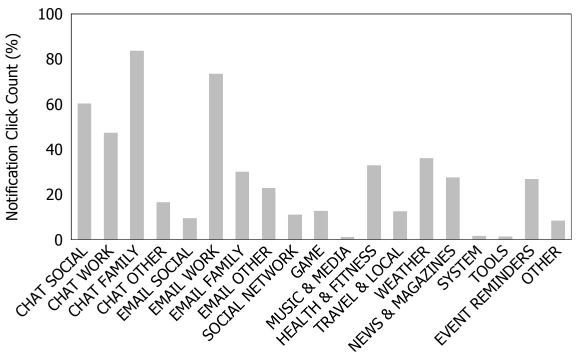


Figure 5. Click count percentages for the notifications of each category.

of the notifications that arrived while the user was in a vehicle were responded to within 10 minutes. At the same time, when the users are still, walking, or running they tend to respond to almost around 60% of the notifications within 10 minutes. It is worth noting that the response rate gradually becomes flat after 10 minutes in all of these three cases. On the other hand, the activity of riding a bike is associated with inopportune moments to deliver a notification.

Impact of Content on Notification Acceptance

In this subsection we analyze the impact of the content on the acceptance of notifications. Since we are interested in the accepted notifications, we classify all the notifications with the response time greater than the threshold time (i.e., 10 minutes) as declined notifications.

In Figure 5 we evaluated the average percentage of the notifications that are accepted in a day for each notification category. The results demonstrate that the notifications from different categories have a varying acceptance rate. The family chat and work email notifications have the highest acceptance rate with around 81% and 77% of the notifications accepted within 10 minutes of arrival time, respectively. On the other hand, notifications of categories such as system, tools, and music and media, have the worst acceptance. Around 10% of the overall notifications in the dataset of labeled notifications belong to the tools category (see Figure 2) but around 99% of the time these notifications were declined by the users. This demonstrates that such notifications almost never provide useful information at the right time to the users and instead become a source of disruption.

It is worth noting that the event reminder notifications might have a low acceptance rate because 1) these notifications provide almost all the information in the notification title; 2) these notifications enable the user to snooze, or delete the reminder through the notification itself without launching the application. For this reason it becomes difficult to capture the response of a user for an event reminder notification.

PREDICTING THE RIGHT TIME FOR A NOTIFICATION

In this section we test our hypothesis: that the acceptance of a mobile notification relies on the type of information in it and the user's context, by analysing the labeled dataset of collected notifications.

Feature	Rank	Average IG
App Name	1	0.251
Notification category	2	0.247
Phone status	3	0.092
Location	4	0.081
Arrival hour	5	0.073
Ringer mode	6	0.056
User's activity	7	0.042
Priority	8	0.026
Alert type	9	0.024
Proximity	10	0.017
Surrounding sound	11	0.003
WiFi connectivity	12	0.001

Table 3. Ranking of features from the NotifyMe dataset.

Data Setup

As the response time of notifications can be very large if they are delivered at the inopportune moments, it contrasts with our objective of finding the opportune moment to deliver a notification that will be handled immediately. Therefore, we label all notifications which were accepted within the threshold response time (i.e., 10 minutes) as "accepted" and the rest as "declined".

Feature Ranking

To understand the value of these features, we rank each feature based on the information gained by adding it for predicting the notification acceptance. We use the *InfoGainAttributeEval* method from WEKA [13] to derive the information gain (IG) each of the attributes brings to the overall classification of a notification as accepted or declined. Table 3 shows the average ranking of the features. The feature evaluation used 10-fold cross validation. Our results show that the name of the application from which a notification is triggered and the notification category are the most important features.

Building Prediction Model

We build individual-based models for predicting notification acceptance by using three different algorithms: Naive Bayes, AdaBoost, and Random Forest. We use two approaches for building prediction models: 1. Data-driven learning that relies on the evidences rather than personal intuitions; 2. User-defined rules that rely on the user's own intuitions.

Data-driven learning

The data-driven learning relies on all the features about the notifications that are collected via the NotifyMe application. As discussed earlier, we derive the category of a notification by using the type of information in it and the recipient's relationship with the sender. However, in order to evaluate the value of using the information type and social circle, we build the prediction models in three ways:

1. without using information type and social circle;
2. using only information type;
3. using information type and social circle.

To build a prediction model in the first and third ways, we simply excluded and included the "category" feature respectively while training the algorithms. However, in order to

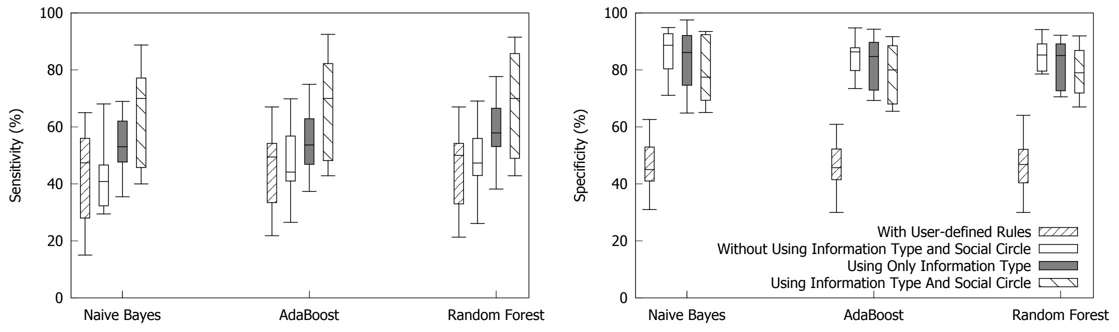


Figure 6. Prediction results of the predictors trained by using 3 different set of features for data-driven learning and user-defined rules.

train a predictor in the second way, we use modified "category" feature that is derived by using only the information type. For example, work email was labeled as email, and social chat was labeled as chat.

User-defined Rules

The user-defined rule-based learning relies on users' responses to the questionnaires triggered by NotifyMe application. We collected 1456 questionnaire responses from the 17 users. The analysis of these responses shows that users are not consistent in defining the rules for the delivery of notifications. Therefore, we used the following 3 features from each questionnaire response (see Table 2) to build a prediction model:

1. *notification category* for which the questionnaire was triggered;
2. *best location* where the user wants to receive notifications with similar content;
3. *best time* when the user wants to receive notifications with similar content.

The features in each questionnaire contributed to a rule for acceptance of a category of notifications. For example, a questionnaire response by a user containing "social chat" as content category, "home and other" as the best location, and "morning" as the best time defines that all social chat notifications that are delivered in the morning when the user is at home will be clicked. Otherwise, the social chat notifications will not be clicked by the user.

Evaluating the Predictors

We evaluate the data-driven prediction models by using the k-fold cross validation approach with the value of k as 10. To evaluate the prediction models for user-define rules, we use all notifications of the respective users. Figure 6 shows the sensitivity and specificity of all the predictors. The sensitivity refers to the proportion of actual positives which are correctly identified (i.e., number of notifications that are correctly predicted as accepted / number of notifications that are actually accepted). On the other hand, the specificity refers to the proportion of actual negatives which are correctly identified (i.e., number of notifications that are correctly predicted as declined / number of notifications that are actually declined).

Our results demonstrate that there is no significant difference in the performance of the three prediction algorithms. The user-defined rules do not perform well as compared to the data-driven learning. The user-defined rule-based predictors only achieve the sensitivity of 55% and specificity of 45%. A possible explanation is that the user preferences change with time. For example, a user might specify that she wants to receive social chats only in the morning, but on the next day she might specify another time period for the same category of notifications. This is a reason why the predictors for user-defined rules perform differently. Moreover, users can define very abstract rules as compared to the complex rules created with the data-driven approach by using numerous other features.

As we trained the data-driven predictors with different features, the results show that the predictor trained with "information type and social circle" indeed outperforms all the other predictors. The predictor trained with "information type and social circle" achieves the sensitivity equal to 70% and the specificity up to 80%. Its specificity remains slightly lower than the specificity of the other two data-driven predictors, but far better than the predictor trained by using user-defined rules. It is because of the trade-off between sensitivity and specificity. Thus, the predictor using the "information type and social circle" feature attains high sensitivity by loosing some specificity. It is worth noting that the user preferences change with time. Therefore, by including the noisy features from the user-defined model, the accuracy of the data-driven predictor will definitely go down.

However, the aim of an interruptibility management system is not only to reduce the disruptive notifications (i.e., high specificity), but also to provide the notifications that are required by the user. This suggests that the use of information type and the recipient's relationship with the sender can boost the performance for predicting the interruptibility. It is worth noting that for some users the sensitivity value went to around 89%. This demonstrates that some people follow their regular behavioral pattern. The predictions for the behavior of such users can be very accurate due to minimal uncertainty in their behavioral patterns.

GENERIC VS PERSONAL BEHAVIORAL MODEL

In this section we compare the performance of the prediction models trained on a user's personal data with a generic prediction model trained on multiple users' data. We build both individual-based models and a generic model for predicting

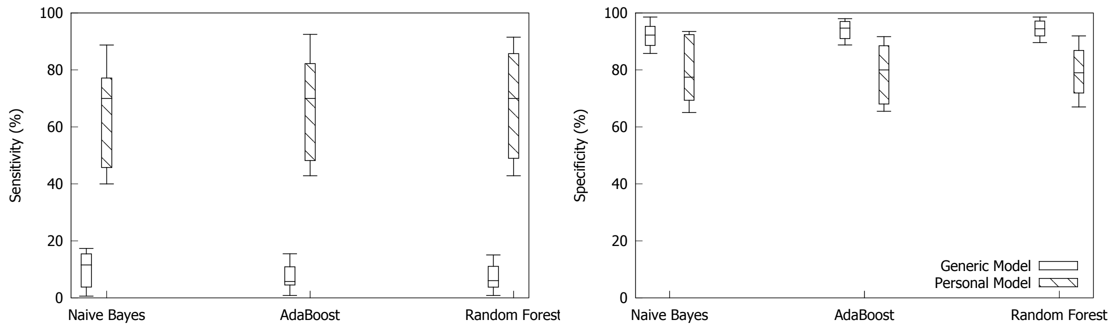


Figure 7. Prediction results of the generic and individual-based models.

notification acceptance by using three different algorithms: NaiveBayes, AdaBoost, and Random Forest. We trained the data-driven models in the same fashion as discussed in the earlier section. We evaluate the individual-based models by using the k-fold cross validation approach and the generic model with the entire set of notifications of each user.

As shown in Figure 7, the individual-based model outperforms the generic model with a very high difference of sensitivity value. At the same time, the generic model achieves extremely high specificity (i.e., around 95%) because it gets trained with most of the rejected notifications in different contexts. Thus, the generic model predicts most of the notifications as declined. An interruptibility management system using such a prediction model will mostly not allow any notification to be triggered. This will be same as switching off the phone which ensures no notification is triggered, but also takes away the benefit of receiving relevant notifications. This demonstrated that a prediction model trained on a user's personal data is more accurate for predicting interruptibility of that user than a generic prediction model trained on multiple users' data.

ONLINE LEARNING

The predictors discussed in the previous section were geared towards batch learning in which the models are trained with the static data. Such an approach has two key drawbacks when used on mobile phones. First, the data becomes available gradually as the new notifications arrive and, therefore, a personalized model cannot be trained until a sufficient amount of data is available. Second, the prediction accuracy can be dramatically reduced when a user changes his/her behavioral pattern. Moreover, data are not usually available for new applications installed by users. For example, when a user starts using a new email client or a social networking application, the model might fail to make any accurate predictions for the notifications triggered by these applications.

To overcome these drawbacks we can use the online learning approach in which the models are regularly trained with the newly available data. Such an approach enables relatively fast learning for new applications (i.e., predictions can be made in the initial days), adaptation to changing or new behavior; and improvement of the performance over time (i.e., the average prediction accuracy can be enhanced gradually as more and

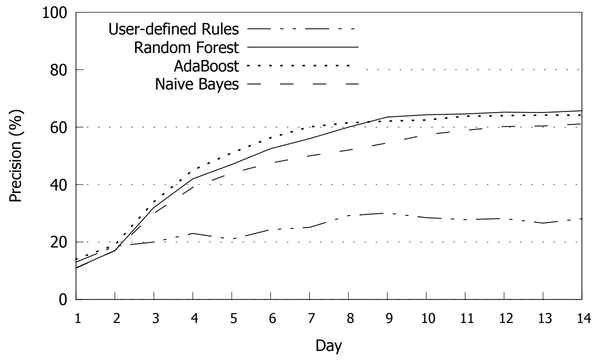


Figure 8. Prediction results of different predictors using online learning approach.

more data becomes available, in case the behavioral patterns do not show strong variations over limited amount of time).

We use our labeled dataset to demonstrate the prediction accuracy of different algorithms by using the online learning approach. We iteratively build all the prediction models with the notification data collected by the end of each day and evaluate these models by using notifications of the following day. For example, on day N a model was built by using notifications from day 1 to N and it was evaluated to predict the acceptance of notifications that arrived on day N+1. Similarly, we train the models for user-defined rules by using questionnaire responses of respective users collected by the end of each day, and evaluate these models by using the notifications of the following day.

We test the data-driven models with three prediction algorithms: Naive Bayes, AdaBoost, and Random Forest. We use user-defined rules as the baseline to evaluate the prediction results of other predictors. As shown in Figure 8, AdaBoost and Random Forest are the best performers. In the initial days, AdaBoost starts makes better predictions as compared to Random Forest. However, the prediction accuracy of AdaBoost stabilizes after seven days. On the other hand, the Naive Bayes algorithm initially shows a prediction accuracy performance comparable to AdaBoost and Random Forest, but its rate of increase in accuracy slowly declines. Overall, all the predictors outperform the user-defined rule mechanism after 2 days. It is worth noting that the best two predictors show the best performance after 9 days of training.

DISCUSSION

According to our measurements, the users in our study receive around 100 notifications each day on average. These notifications arrive at different times of the day without any knowledge about recipients' willingness to get interrupted. Therefore, such notifications have a potential to convert the smartphone from an information-awareness device to an information-overloading one. At the same time, our results show that the response to a notification can be predicted. The user's predicted response can be used to ensure the delivery of notifications at opportune moments.

To predict the user response to a notification we need to build a user's behavioral model and train it with the notifications of various categories that arrive in different contexts. It is possible to capture the user's context via a smartphone's embedded sensors. On the other hand, to categorize notifications we need the knowledge about the information contained in them. Some notifications can be classified by using the knowledge about the category of applications which triggered them. For example, an application of news category is very likely to trigger notifications containing news related information.

Our results show that around 70% of the notifications that a user receives belong to email and chat categories. Thus, a further subcategorization of these two categories is needed. As proposed in this study, we can use the social circles of users to sub-categorize email and chat notifications. For example, a chat notification can be sub-categorized as "social chat" if the sender of the notification belongs to the friend circle defined by the user. Communication applications (e.g., email and chat) allow the users to map their social ties in the social circles defined by them. One possibility is that these circles can be used by an application to categorize email and chat notifications. However, the users do not create social circles on every communication applications and also the reverse is true that not all communication applications facilitate users to create social circles. This raises an interesting question about how to maintain the knowledge about users' social circle that can be used to sub-categorize the chat and email notifications.

In a recent study [18], authors argue that the operating system of a mobile device should be responsible for managing and delivering notifications at the opportune moments. Inspired by the authors argument, we suggest a solution to the problem, posed above, to maintain the knowledge about a users' social circle. Since the operating system has access to all the information of every application, there should be a system service (such as Google Play Service for Android OS) that can categorize all notifications with respect to the application which triggered it and an aggregated social circle of the user generated by merging social circles from different applications (especially from the social networking and calendar applications). Such a service can monitor the user's response for all notification categories in different contexts. Thus, there will be more data available to build a richer model of the user's behavior as compared to the models created by the applications themselves. Also, we believe that this service can provide an efficient solution because we only need to train a single but rich behavioral model of a user.

LIMITATIONS

In this work we consider two cases of user's response to a notification – accept (i.e., when a user clicks on the notification to read the information contained in it) and decline (i.e., when a user ignores or does not answer the notification up to a certain time delay). However, it is possible that some notifications were misclassified as declined because they might be actually attended by the users on another device or they might be read and dismissed by the users because they do not require further actions. Thus, the prediction model is trained by means of incorrect data.

We also identify a series of aspects that deserve further investigation, probably also by means of experience sampling techniques. One issue is related to multiple notifications from the same applications (usually referred to as "stacked" notifications). How does a user react when the notifications are stacked by an application? Does a user click/ignore it because all notifications are irrelevant or because a notification (perhaps the latest one) from the stack is important/not important? Another important aspect is related to the influence of the user's co-location with other people when a notification is received. For example, a certain user sitting in a coffee shop alone might be willing to accept a notification. But, when the same user is sitting with colleagues and discussing a project, she might not accept any notification.

RELATED WORK

Sahami et al. [28] ran a large-scale study to understand how users perceive mobile notifications. Their results show that the users deal with many notifications each day, and most of the notifications are viewed within a few minutes of arrival. Additionally, by collecting the subjective feedback from the mobile users, the authors show that users assign different importance to notifications triggered by application from different categories. In a recent study, Pielot et al. [25] show that the personal communication notifications are responded to quickly because of the social pressure and the exchange of time critical information by the communication applications. Although, these notifications make the users feel connected with their social links, the increasing number of such notifications also becomes a source of negative emotions and stress. These results demonstrate that the users are not always interested in the information pushed by the proactive services. Thus, it becomes necessary for such services to push only the right information at the right time. At the same time, as Iqbal et al. [16] suggest, users are willing to tolerate some disruption in return for receiving notifications that contain valuable information.

Most of the work on mobile interruptibility management focuses on the user's context for inferring opportune moments to interrupt [10, 15, 16]. Ho and Intille [15] explore the delivery of notifications at the transition between physical activities. Their results suggest that notifications might be considered more positively when delivered between two physical activities, such as sitting and walking. Similarly, Fischer et al. [10] investigate whether the transitions between the user's interaction with the mobile phone are opportune to deliver notifications. They conducted an experience sampling study in which the notifications asked users to provide feedback about their context (for instance, taking a photo of the surroundings) during different phases of their interaction with mobile phones. They found that the participants react faster to notifications when they are delivered immediately at the end of a task such as after finishing a phone call or reading a text message. In a recent study, Pielot et al. [26] identified that the shared indicators of availability (such as the last time the user has been online) are weak predictors of users' responsiveness for notifications. The authors show that features, such as the last notification response, phone's ringer mode and user's proximity to the screen, are good predictors of a notification's response time, and can predict whether a notification will be seen by the recipient within a few minutes with 70.6% accuracy. These studies remain limited to detect generic moments to interrupt users by notifications and fail to provide a complete picture of user's willingness to accept a notification. This is due to the fact that they ignore the role of content for modeling interruptibility. In this study, instead, we will discuss the design of a *content-driven* mechanism for predicting an opportune moment to deliver a notification.

A handful of previous studies have incorporated aspects of content for studying interruptibility [12, 29, 11]. Grandhi and Jones [12] propose a theoretical framework for interruptibility management that takes into account who the interruption is from and what it is about, in a desktop-based setting. The authors do not provide evidence to show how the framework can work in a mobile environment. Speier et al. [29] use the relevance of the notification content with the ongoing desktop activity to predict the cost of interruption caused by the delivered information. Such an approach can work in desktop environments where the users' context is not very dynamic. The pervasive nature of mobile phones brings in various additional contextual features, in particular the fact that the devices are used, essentially, everywhere and potentially at any time of the day. Thus, it becomes important to consider the relevance of *content within the context*.

On the other hand, Fischer et al. [11] investigate the effects of content and the time of delivery of a mobile notification on users' response. The authors show that a notification's response is influenced by the user's general interest in the notification content, entertainment value perceived in it and actions required by the notification, but not the time of delivery. However, first, the authors do not fully justify the ecological validity of their approach because the notifications delivered to the users were not real world notifications but were generated by their own system; second, they consider the general interest of the users and entertainment value in the information, but we argue that the value of such factors are perceived differently by the users in different contexts. Our intelligent notification mechanism not only overcomes such limitations by considering user's behavior about "what (type of information)" and "where (user's context)" to receive mobile notifications, but also uses the social circles to exploit the recipient's relationship with the sender.

To the best of our knowledge there is no mobile interruption study that considers both of these two factors together to predict opportune moments for delivering mobile notifications. Moreover, this study uses *in-the-wild* notifications for the first time to model interruptibility; indeed, existing work is based on synthetic notifications or desktop-based experiments.

CONCLUSIONS AND FUTURE WORK

Mobile notifications provide an effortless way to enable users to be aware of newly available information in quasi real-time. However, notifications arrive at inappropriate moments or carry irrelevant content. In this paper we have presented an analysis of the impact of content on the acceptance of mobile notifications. The results of the analysis have been used to develop a novel machine learning approach that predicts the acceptance of a notification by using its content and the context in which it is delivered. Such an approach can be used by an interruptibility management system to ensure that the right information is delivered at the right time.

We have collected notifications “in-the-wild" and classified them according to the information contained in them. Our results have shown that a user's activity can impact the time delay in the response to a notification. We have evaluated the average percentage of the notifications that are accepted in a day for each notification category. Our results have shown that the chat notifications, where the sender is a family member or a relative of the user, have the highest acceptance rate. Overall, the acceptance value of notifications vary for each category. By using the collected real-world notifications we have designed, developed and tested a series of predictors based on state-of-the-art machine learning. We have shown that the acceptance of a notification within 10 minutes from its arrival time can be predicted with an average sensitivity of 70% and a specificity of 80%. For some users the sensitivity can go up to 89%. Our approach outperforms the user-defined rules for delivering notifications on their mobile phones. Finally, we have discussed the implementation of an online predictor in order to understand the required training period for successful prediction.

We consider this work an initial step towards the implementation an effective and efficient component for notification management. We believe that the accuracy of the prediction model will increase by considering fine-grained categories that can be obtained by classifying the notification content using natural language processing techniques. Another orthogonal issue that will play a fundamental role in the development of these technology is privacy: indeed, the implementation of the proposed algorithms involve the analysis of user-generated content and of the social relationships of the users. Our goal is to explore ways for performing advanced processing on the phones in order to minimize the sharing of personal information with third-party entities.

ACKNOWLEDGEMENTS

This work was supported through the EPSRC grants “UBhave: ubiquitous and social computing for positive behaviour change" (EP/I032673/1) and “Trajectories of Depression: Investigating the Correlation between Human Mobility Patterns and Mental Health Problems by means of Smartphones" (EP/L006340/1).

REFERENCES

1. Android's Notification Listener Service. `http://developer.android.com/reference/android/service/notification/NotificationListenerService.html`.
2. Google's Activity Recognition Application. `http://developer.android.com/training/location/activity-recognition.htmll`.
3. Adamczyk, P. D., and Bailey, B. P. If not now, when?: the effects of interruption at different moments within task execution. In *CHI'04* (April 2004).
4. Bailey, B. P., and Konstan, J. A. 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.
5. Bailey, B. P., Konstan, J. A., and Carlis, J. V. Measuring the effects of interruptions on task performance in the user interface. In *SMC'00* (October 2000).
6. Clark, H. H. *Using language*, vol. 1996. Cambridge University Press Cambridge, 1996.
7. Consolvo, S., McDonald, D. W., Toscos, T., Chen, M. Y., Froehlich, J., Harrison, B., Klasnja, P., LaMarca, A., LeGrand, L., Libby, R., et al. Activity Sensing in the Wild: a Field Trial of UbiFit Garden. In *CHI'08* (April 2008).
8. Cutrell, E., Czerwinski, M., and Horvitz, E. Notification, disruption, and memory: Effects of messaging interruptions on memory and performance. In *Interact'01* (July 2001).
9. Czerwinski, M., Cutrell, E., and Horvitz, E. Instant messaging and interruption: Influence of task type on performance. In *OZCHI'00* (November 2000).
10. Fischer, J. E., Greenhalgh, C., and Benford, S. Investigating episodes of mobile phone activity as indicators of opportune moments to deliver notifications. In *MobileHCI'11* (September 2011).
11. Fischer, J. E., Yee, N., Bellotti, V., Good, N., Benford, S., and Greenhalgh, C. Effects of content and time of delivery on receptivity to mobile interruptions. In *MobileHCI'10* (September 2010).
12. Grandhi, S. A., and Jones, Q. Conceptualizing interpersonal interruption management: A theoretical framework and research program. In *HICSS'09* (January 2009).
13. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter 11*, 1 (2009), 10–18.
14. Heinonen, K., and Strandvik, T. Consumer responsiveness to mobile marketing. In *Stockholm Mobility Roundtable* (May 2003).
15. Ho, J., and Intille, S. S. Using Context-Aware Computing to Reduce the Perceived Burden of Interruptions from Mobile Devices. In *CHI'05* (April 2005).
16. Iqbal, S. T., and Horvitz, E. Notifications and awareness: a field study of alert usage and preferences. In *CSCW'10* (February 2010).
17. Lathia, N., Rachuri, K. K., Mascolo, C., and Roussos, G. Open Source Smartphone Libraries for Computational Social Science. In *MCSS'13* (September 2013).
18. Lee, K., Flinn, J., and Noble, B. The case for operating system management of user attention. In *HotMobile'15* (February 2015).
19. Lu, H., Pan, W., Lane, N. D., Choudhury, T., and Campbell, A. T. SoundSense: Scalable Sound Sensing for People-centric Applications on Mobile Phones. In *MobiSys'09* (Krakow, Poland, June 2009).
20. McFarlane, D. C., and Latorella, K. A. The scope and importance of human interruption in human-computer interaction design. *Human-Computer Interaction 17*, 1 (2002), 1–61.
21. Mehrotra, A., Pejovic, V., and Musolesi, M. SenSocial: A Middleware for Integrating Online Social Networks and Mobile Sensing Data Streams. In *Middleware'14* (December 2014).
22. Miyata, Y., and Norman, D. A. Psychological issues in support of multiple activities. *User centered system design: New perspectives on human-computer interaction* (1986), 265–284.
23. Monk, C. A., Boehm-Davis, D. A., and Trafton, J. G. The attentional costs of interrupting task performance at various stages. In *HFES'02* (October 2002).
24. Pejovic, V., and Musolesi, M. Interruptme: designing intelligent prompting mechanisms for pervasive applications. In *UbiComp'14* (Seattle, WA, USA, September 2014).
25. Pielot, M., Church, K., and de Oliveira, R. An in-situ study of mobile phone notifications. In *MobileHCI'14* (September 2014).
26. Pielot, M., de Oliveira, R., Kwak, H., and Oliver, N. Didn't you see my message?: predicting attentiveness to mobile instant messages. In *CHI'14* (April 2014).
27. Rettie, R., Grandcolas, U., and Deakins, B. Text Message Advertising: Response Rates and Branding Effects. *Journal of Targeting, Measurement and Analysis for Marketing 13*, 4 (2005), 304–312.
28. Sahami Shirazi, A., Henze, N., Dingler, T., Pielot, M., Weber, D., and Schmidt, A. Large-scale assessment of mobile notifications. In *CHI'14* (April 2014).
29. Speier, C., Vessey, I., and Valacich, J. S. The effects of interruptions, task complexity, and information presentation on computer-supported decision-making performance. *Decision Sciences 34*, 4 (2003), 771–797.

30. Weiser, M., and Brown, J. S. *The coming age of calm technology*. Springer, 1997.

31. Xu, Y., Lin, M., Lu, H., Cardone, G., Lane, N., Chen, Z., Campbell, A., and Choudhury, T. Preference, context and communities: A multi-faceted approach to predicting smartphone app usage patterns. In *ISWC'13* (September 2013).

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