Challenges in Managing Multi-Device Notifications

Abhinav Mehrotra
University of Birmingham
University College London
United Kingdom
a.mehrotra@cs.bham.ac.uk

Jo Vermeulen
University of Birmingham
United Kingdom
j.vermeulen@cs.bham.ac.uk

Robert Hendley
University of Birmingham
United Kingdom
r.j.hendley@cs.bham.ac.uk

Mirco Musolesi
University College London
United Kingdom
m.musolesi@ucl.ac.uk

Submitted to the *Cross-Surface '15* workshop, in conjunction with ACM *ITS'15*. November 15, Funchal, Madeira, Portugal.

Abstract
Many applications and online services on our mobile phones notify us about messages, emails, or status updates. These notifications can be disruptive. People currently use multiple mobile and wearable devices that can run the same application or service simultaneously. This aggravates the issue of disruptive notifications. For example, a new email notification can trigger vibrations or sounds on a tablet, laptop, smartphone and smartwatch at the same time. Additionally, notifications can be sent from one device to another (e.g., an internet-connected thermostat that notifies a user on their smartphone). In this position paper, we discuss the key challenge of handling and managing multi-device notifications. We outline open issues in addressing this problem and discuss opportunities for future work.

Author Keywords
Notifications, Interruptibility, Mobile Sensing, Multi-Device Interaction, Wearables

ACM Classification Keywords
H.5.m. [Information Interfaces and Presentation (e.g. HCI)]

Figure 1: Mark is chatting with Susan via an instant messaging application. When Susan replies, he receives multiple notifications on his smartwatch, laptop and smartphone (sketch by Lindsay MacDonald).

Introduction

People currently tend to own and use multiple personal devices, including smartphones, tablets, laptops, wearables and internet-connected smart devices (e.g., smart thermostats or smart scales). While the perpetual availability of information through these devices and increased connectivity is generally considered to be beneficial, people also get interrupted through constant notifications on their devices. One critical problem reported by early adopters of smartwatches is being constantly interrupted by notifications. Despite this, early adopters also report the ability to unobtrusively receive information in social situations as an advantage of wearing a smartwatch [5]. Another study found that some smartwatch compulsively check their watches and even check their bare wrists when they were not wearing their devices [2]. Some reported regularly feeling a phantom buzz, notifying them of an imagined incoming message.

Another issue is that our personal devices tend not to be aware of each other. Notifications often trigger vibrations or sounds on all of our devices at the same time. In Figure 1, Mark is discussing a problem with his colleague Susan via an instant messaging application on his laptop. Mark has the same instant messaging application installed on both his smartwatch and smartphone. When Susan replies to Mark's message, notifications arrive on all three devices to alert Mark of Susan's incoming message.

This demonstrates the need for more intelligent multi-device interruption management. If applications are aware of a user's different devices, they could determine the best device to use to alert the user. In this case, the notification could just have appeared on Mark's laptop. If Mark was walking around the office to get a coffee, however, the notification could have been delivered to his smartwatch instead.

In order to achieve this kind of flexibility, an intelligent notification approach should not only infer the user's receptivity to notifications in a particular context, but also learn how the user interacts with notifications on each of their devices in that context. In other words, multi-device applications should consider *when* and *whether* to deliver a notification and additionally *where* (i.e., on which device) this notification should be delivered. Multi-device interruption management can benefit applications that trigger push notifications by reducing interruptions from multiple simultaneous notifications.

Related Work

Previous studies have found that initiating interactions at inopportune moments can cause interruptions [8]. These interruptions can adversely affect task completion time [6], error rate [4], and can influence people's affective state [3]. Additionally, people may feel anxious about missing out when they cannot check their devices for incoming notifications [9]. Studies suggest that some people tend to receive hundreds of notifications per day [7].

Researchers have proposed different mechanisms for managing interruptibility. Adamczyk and Bailey [1] propose a method to predict interruption timings based on a user's cognitive load during the task execution. Pejovic et al. [10] use sensed information (e.g., the location, time of day, activity) to infer opportune moments to deliver notifications. Mehrotra et al. [7] show that machine learning classifiers lead to a more accurate prediction of a user's interruptibility when trained with both the content of a notification and the user's context. All of these studies focus on interruptions caused by a single device and do not consider multi-device interruptions.

Challenges

In this section, we discuss a number of research challenges for multi-device interruption management.

Deciding Where to Interrupt?

An important question for intelligent multi-device notification management is to decide *where*, *when*, *whether* and *how* to interrupt the user. A naïve approach would be to detect when people are actively using one of their devices (e.g., by monitoring input events) and direct all notifications only to that particular device. However, there are other aspects that are important to consider in determining the most suitable device to use. For example, some devices can be more suitable to use in a certain context than others (e.g., while driving).

Inferring Engagement with a Device
Different techniques can be used to determine whether people are actively using their device. Input events can be monitored to infer how actively engaged the user is with a device (e.g., key presses on a laptop).

Smartphones and smartwatches tend to turn off their displays after a certain period of inactivity to optimise battery usage. The status of the display (on or off) can be tracked and used as a simple measure of engagement.

Gaze tracking can be an rich, albeit more complex and resource-intensive, way of measuring engagement by tracking whether the user is actually looking at the device. Some Android smartphones already feature built-in gaze trackers. One application of gaze tracking is to verify that users have seen a certain notification without requiring them to interact with it first (e.g., clicking on an email to mark it as unread).

Predicting Which Device Notifications Will Be Handled On
Daily use of personal devices together with the increasing popularity of wearables and the presence of connected embedded sensors in many artefacts and the fabric of cities themselves (i.e., the "Internet of Things") allows us to extract and model some inherent patterns of human behaviour. More specifically, user behaviour in specific sensed contexts observed over a period of time can be used to build predictive models. In order to predict the device on which the user will handle a notification in a given context, an application could learn the patterns of the user's receptivity to information on different devices in different situations.

Context prediction has been investigated in the past for forecasting future locations of users [12], communication patterns [11], and interruptibility [10, 7]. Similar techniques can be used to construct a machine learning model that learns the patterns of the user's interaction with notifications on different devices in different contexts. The model would gradually start making sensible predictions about the device on which the user will handle a certain notification in their current context.

A key challenge is to be able to train the system over multiple devices. Certain devices might be used rarely by a user in a certain context and this can make the learning task very challenging. There is a fundamental problem related to bootstrapping the learning components. A possible approach is to adopt models extracted over multiple users and refine these, for example, using Bayesian approaches.

Privacy and Shared Devices

People might be using devices that can be seen by others (e.g., notifications appearing while giving a presentation). In order to maintain privacy, personal notifications might have to be suppressed on a particular device, even though it is the active one. Additionally, some devices such as tablets might be shared between family members.

Providing User Control

Another important challenge is providing a balance between automatic notification routing and end-user control over the dispatch of notifications to certain devices in a particular context. There are limitations to what can be correctly sensed, which means notification routing algorithms will inevitably make inappropriate judgements. Users could configure rules to route certain types of notifications to certain devices, or the application could defer to the user's judgement when it cannot infer which device is the most appropriate. An important consideration is that asking the user where to deliver notifications can again cause interruptions.

Conclusion

Cross-device interaction enables people to use different devices together and have access to information and complete tasks on several devices. A key challenge in this is interruption overload. All these devices and applications are constantly competing for the user's attention, with several devices that buzz or beep at the same time. In this paper, we have discussed opportunities and challenges for intelligent multi-device notifications. In order to address the issue of multi-device interruptions, applications should be able infer the user's engagement with different devices and learn the patterns of the user's receptivity to information on different devices in different contexts. A key issue in designing multi-device notification management is training the machine learning components, given the fact that particular interactions with certain types of notifications may rarely occur.

References

[1] Adamczyk, P. D., and Bailey, B. P. If not now, when?: the effects of interruption at different moments within task execution. In *CHI'04* (2004).

[2] Andre Spicer. Apple Watch and the Internet of me, Mar. 2015. http://www.straitstimes.com/opinion/apple-watch-and-the-internet-of-me.

[3] 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.

[4] 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* (2000).

[5] Cecchinato, M. E., Cox, A. L., and Bird, J. Smartwatches: The good, the bad and the ugly? In *CHI EA '15* (2015), 2133–2138.

[6] Czerwinski, M., Cutrell, E., and Horvitz, E. Instant messaging and interruption: Influence of task type on performance. In *OZCHI'00* (2000).

[7] Mehrotra, A., Musolesi, M., Hendley, R., and Pejovic, V. Designing Content-driven Intelligent Notification Mechanisms for Mobile Applications. In *UbiComp'15* (2015).

[8] 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.

[9] Oulasvirta, A., Rattenbury, T., Ma, L., and Raita, E. Habits make smartphone use more pervasive. *Personal and Ubiquitous Computing 16*, 1 (2012), 105–114.

[10] Pejovic, V., and Musolesi, M. InterruptMe: designing intelligent prompting mechanisms for pervasive applications. In *UbiComp'14* (2014).

[11] Pielot, M. Large-scale evaluation of call-availability prediction. In *UbiComp'14* (2014).

[12] Scellato, S., Musolesi, M., Mascolo, C., Latora, V., and A., C. Nextplace: A spatio-temporal prediction framework for pervasive systems. In *Pervasive'11* (2011).