Towards Smart Notifications using Research in the Large

Dominik Weber
VIS, University of Stuttgart
Stuttgart, Germany
dominik.weber@vis.uni-stuttgart.de

Alireza Sahami Shirazi[a]
Yahoo Labs
Sunnyvale, USA
alireza@yahoo-inc.com

Niels Henze
VIS, University of Stuttgart
Stuttgart, Germany
niels.henze@vis.uni-stuttgart.de

[a]The majority of the work has been conducted while he was a researcher at the University of Stuttgart.

MobileHCI '15 Adjunct, August 25-28, 2015, Copenhagen, Denmark
ACM 978-1-4503-3653-6/15/08.
http://dx.doi.org/10.1145/2786567.2794334

Abstract
Notifications are a core function of current smart devices. They inform users about a variety of events, such as new messages, comments on social networks posts or application updates. As such, notifications are the main mechanisms to proactively communicate with the user. Focusing on individual device types such as PCs and smartphones, previous work showed that notifications can be distractive and disruptive. The ongoing wave of smart devices makes it possible to reach the user through multiple devices at once – amplifying the effects of notifications. What is missing is an understanding of notifications in a multi-device environment to enable the smart management of notifications across devices. In this paper we present a system that is able to share notifications across smartphones, tablets, PCs, and smart TVs. It can further reach users through connected devices such as smart watches and smart glasses. The system currently distributes 5.3 million notifications by almost 30,000 users every day. It is not only intended to provide a holistic notification mechanism but also serves us to conduct large scale user studies to gain a deeper understanding of notifications in a multi-device environment.

Author Keywords
Notification; Smart Devices; Smart Notification

Introduction and Background

Notifications are current smart devices' main mechanism to proactively inform users. Notifications are used to inform about new messages, to announce upcoming events and to highlight changes of the system's status. With the growing number of smart devices, including smartphones, smart watches and smart TVs, the number of systems that simultaneously aim to reach the user is steadily increasing. Today, a single mail, for example, can cause notifications on half a dozen devices at the same time. While notifications are one of the core mechanisms that make current smart devices 'smart' they can also have negative effects on the user. They can be disruptive and distract from other tasks. To find a balance between keeping the user informed and confine the negative effects it is necessary to find means to coordinate the delivery of notifications across device boundaries in an intelligent way.

Previous work on understanding the effects of notifications and how users interact with them mainly focused on individual types of devices and notifications. Iqbal and Horvitz, for example, investigated the effect of notifications about incoming mails on desktop computers [3] and Pielot et al. investigated the effect of notifications on mobile devices [4]. What is missing is not only a mechanism to coordinate the distribution of notifications across devices. What is also missing are insights about the effects of notification in multi-device environments. It is, for example, even unclear how a user should be informed about an incoming message if multiple devices are available. Notifications could be shown on all devices a user has access to, only on devices in the user's vicinity, or only on the nearest device. Notifications could further be delayed to reduce the frequency notification events or notifications could be distributed across devices depending on their importance.

All major mobile platform providers, including Apple, Google, and Microsoft, recently started to take the first steps towards a holistic management of notifications. Apple, for example, unified the notification API across their mobile and desktop platforms. Academia, however, recognized the need for ubiquitous notification management earlier. Arlein et al., for example, proposed a system that can distribute notification across PCs, phones, PDAs and other devices already in 2008 [1]. Previous work, however, either focused on systems for ubiquitous notifications without evaluating the effect or understanding the effects of notifications for specific device types. What is missing are systems that enable to study the effects of notifications in multi-device environments.

In this paper we present a system to coordinate the distribution of notifications across devices and at the same time to learn about the effects of notifications following a research in the large approach [2]. We developed the system based on our previous work in which we used an early version of to conduct a large-scale assessment of mobile notifications [5]. Our current system is able to distribute notifications across smartphones, tablets, PCs, and smart TVs. Furthermore, notifications can reach the users through connected devices including smart watches and smart glasses. With almost 30,000 daily users Desktop Notifications is also the ideal platform to learn about notifications in multi-device environments.

In the following we present the design of the current version of Desktop Notifications, a system to distribute notification across smart devices. Afterwards, we describe our current users and the devices they use. Finally, we highlight opportunities for further extensions of the system and the opportunities of the system for future studies.

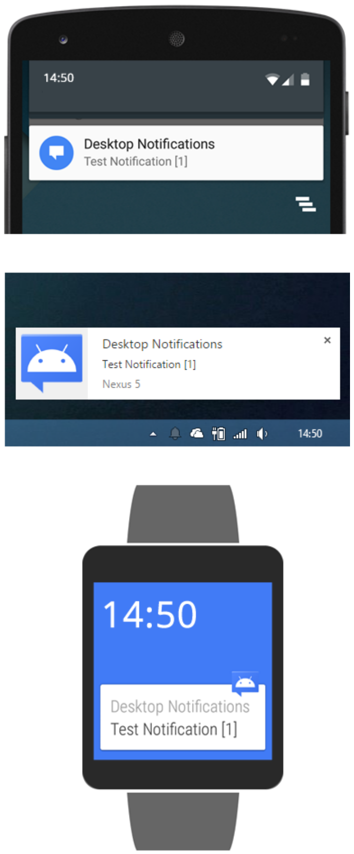

Figure 1: The same exemplary notification shown on an Android smartphone (top), desktop PC (middle) and Android Wear smart watch (bottom).

Architecture

To distribute notifications across multiple smart devices we developed client applications for every type of device. For example, we developed an Android app for Android-based devices. The app automatically detects whether the Android device is a smartphone or a tablet. On the desktop we developed extensions for the Mozilla Firefox and Google Chrome web browsers. Both browsers are available for all major desktop operating systems (Microsoft Windows, Apple OS X and Linux) which enables us to reach a large audience. Figure 1 shows an exemplary notification that was distributed to a smartphone, desktop PC and smart watch. We published the Android app to the Google Play Store and the browser extensions to the corresponding extension platforms. The installation from these platforms is easy, requiring at most two clicks to install a client application. Furthermore, all app stores allow automatic updates, thus distributing new versions of the client applications is painless.

In addition to the client applications we created a server application that keeps track of the user-device-relations. On the first run the client applications authenticate the user via OAuth, an open standard for authentication. The device is then registered on the server along with the user's account ID. Because this information is stored on the server, the user's devices do not have to know about each other. The client applications communicate with the server application via a bi-directional channel. This communication channel does not necessarily have to be persistent, as long as the clients can send data to the server at any time and vice versa. Figure 2 shows the central server component and multiple types of smart devices that are connected to it. Wearable devices, like smart watches, often use Bluetooth to connect to other devices. Because of this, smart watches do not have a direct connection to the server and use a smartphone as a bridge instead. Pushing messages to mobile devices is a common problem. Mobile devices are limited by their battery life. Keeping the device awake to establish a persistent connection should be avoided. One solution to the problem is polling – periodically waking the device and asking the server for new notifications. However the frequency of the polling is a tradeoff between how often the device is active and how fast new data is received from the server. Polling also tends to create lots of 'empty' requests and the data has to be stored on the server at least temporally. In the case of notifications even a ten second delay caused by the polling might be too long. To avoid developers having to solve this problem themselves all major mobile platform providers offer a highly optimized push message service to push messages from a server to devices. For Android Google provides the free Google Cloud Messaging service. We first push messages from our server application to the push message service, which in return sends the message to the devices. Despite this additional step, the messages are typically delivered within one second. The client applications on the devices use the received messages to display new or remove existing notifications.

Currently the source of notifications is limited to Android-based devices. The reason for this is that only Android offers a notification API that allows an app to listen for newly created and removed notifications by other apps, given the user permits the access to the notifications. Our Android app uses this API to listen for all changes regarding to notifications and forwards information like the name of the app that created a notification, the actual content and additional meta data like the time of creation to the server. In return, the server broadcasts this data to all other devices owned by

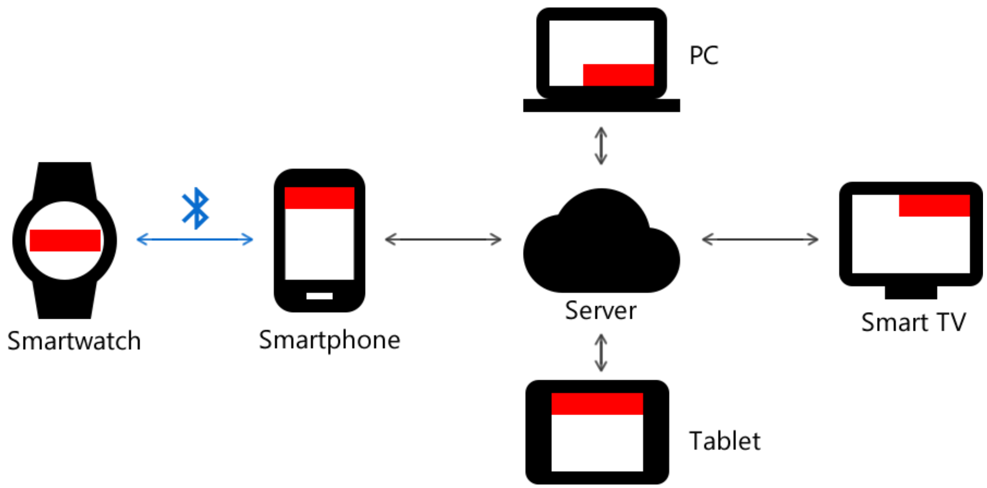

Figure 2: Our system distributes notifications across multiple types of devices. A central server application keeps track of the user-device-relations and allows broadcasting the notifications to all devices of a user.

the user. Information regarding the removal of notifications is broadcasted in the same manner. The distributed notifications stay on every device until they are dismissed by the user or programmatically removed by the app that created the notification in the first place.

Usage Statistics

We released the first version of Desktop Notifications in November 2012. In this version the notifications from an Android device were sent to a server and an extension for the Google Chrome browser periodically polled the server for new notifications. It was only possible to send data in one direction. The Android app was not able to receive data from the server, thus displaying a notification from one Android device on another Android device was not possible. Furthermore the user had to dismiss the notification on both the Android device and the desktop. Despite the limitations the app was featured in a popular German blog in January 2013 and was subsequently picked up by other blogs and news sites around the world. By the end of 2013 the Android app was downloaded over 115,000 times and by the end of 2014 over 300,000 times. Until November 2014 the architecture of the system did not change compared to the first version that we released almost two years prior. In a coordinated effort we released updates to all parts of the system to distribute the architecture that we described in the previous section.

The update required the users to pair all devices anew and old versions of the app ceased to function. Six months after the update the app has been installed 372,280 times and is currently installed on over 62,495 Android devices that belong to 56,196 unique users according to the Google Play Store. The total number of devices is divided

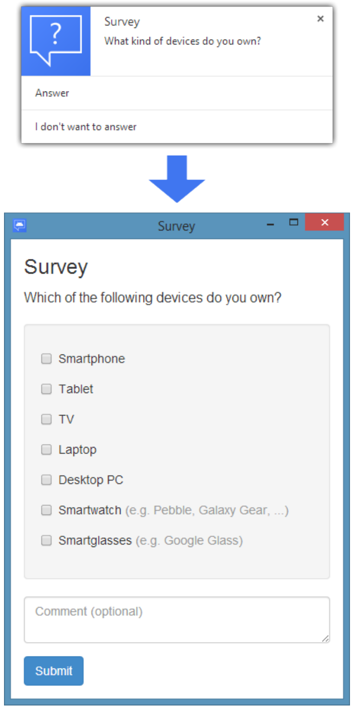

Figure 3: We asked the users of the Google Chrome extension to participate in a survey. Within 24 hours we were able to distribute the survey to thousands of users.

in 55,772 (89.24%) smartphones and 6,723 (10.76%) tablets. The most popular countries in the order of popularity are the United States, Germany, Brazil, the United Kingdom and India. The Android app has been rated 9,554 times with an average rating of 4.25 out of 5 stars. Furthermore the app received over 1,900 comments. In the comments users praise and rant about the app, suggest features and report bugs.

On the desktop, the extension for the Google Chrome browser is installed on 114,044 devices according to the Chrome Web Store. The browser extension for the Mozilla Firefox web browser is installed on 12,098 devices according to the statistics provided by Mozilla. The discrepancy regarding user count between the browser extensions and the Android app indicates, that many users might install the browser extension out of curiosity but fail to install the Android app, or simply do not bother to remove the extension from browser after they decide to no longer use the app. The Chrome extension has been rated 714 times with an average rating of 4.21 out of 5 stars and the Firefox extension has been rated 23 times with an average rating of 4.62 out of 5 stars.

The server application currently handles approximately 5.3 million notifications from almost 30,000 users per day. From our log files we can see that the notifications are created by almost 9,000 distinct Android applications. It is unlikely that we would be able to record such a rich diversity in a controlled lab study.

Gaining Insights

Besides analyzing the notifications that are processed by our server, we can use the system to quickly distribute surveys to users. For example we added a survey to the Google Chrome extension that asked the users what kind of devices they own. We included the survey in an update for the extension. Google Chrome automatically takes care of updating extensions and an update typically is completely rolled out within a day. As shown in Figure 3, we prompted the users to take part in the survey by showing a notification once a day. The notification was shown repeatedly until the user either took part in the survey or actively declined to take part in it. If the user decided to participate, a pop-up window was shown with a list of predefined possible answers and an optional comment field. The survey was shown for four weeks and subsequently removed with another update to the extension. In this timespan we were able to collect answers from 6,779 users.

As the server knows which types of devices the users own, we can target subsets of users. In another experiment we updated both the Android app and Chrome extension and asked users that own a smartphone, tablet and desktop PC to rate the usefulness of notifications on the current device. The results show that users value notifications related to communication regardless of the device type, whereas other types of notifications showed mixed results.

Furthermore, we gain insights into the user's preferences by analyzing the settings of the client applications. In the Android app, for example, we allow the users to disable distributing notifications for certain apps. Additionally, users can mark certain apps as private. Notifications from private apps are still distributed across all devices, however without the actual content of the notification. Analyzing these settings showed that most apps marked as private are related to personal communication, whereas blocked apps are typically system notifications that inform the user about available updates, completed downloads or results of malware detection apps.

Conclusion and Future Work

In this paper we presented a system we developed to coordinate the distribution of notifications across devices and at the same time to learn about the effects of notifications following a research in the large approach [2]. We presented the architecture of our system, which consists of multiple client applications and a server component. The Android-based client application is able to detect all notifications created on a device. It sends information regarding new and removed notifications to a central server, which in turn broadcasts it to all devices that belong to a user. The service currently handles 5.3 million notifications from almost 30,000 users per day.

For future work we want to explore how we can create a truly smart notification system. It should be explored if it is possible to automatically determine the most suitable device or subset of devices for a given context and notification instead of simply broadcasting the notification to all devices. Prior work has already investigated how mobile phones can infer where they are kept [6]. We aim to build upon this work by combining this information with knowledge about characteristics of notifications. Furthermore, many devices support multiple notification modalities to alert the user. Again, based on the current context and importance of the notification, the modalities to alert the user should be used in degrees, ranging from subtle to intrusive. In summary, it should be explored when, on which devices and by what means the user should be notified.

Acknowledgments: This work is supported by the German ministry of education and research (BMBF) within the DAAN project (13N13481) and by the DFG within the SimTech Cluster of Excellence (EXC 310/2).

References

[1] Arlein, R. M., Betge-Brezetz, S., and Ensor, J. R. Adaptive notification framework for converged environments. *Bell Labs Technical Journal 13*, 2 (2008).

[2] Henze, N., Pielot, M., Schinke, T., and Boll, S. My app is an experiment: Experience from user studies in mobile app stores. *IJMHCI* (2012).

[3] Iqbal, S. T., and Horvitz, E. Notifications and awareness: a field study of alert usage and preferences. In *Proc. CSCW* (2010).

[4] Pielot, M., Church, K., and de Oliveira, R. An in-situ study of mobile phone notifications. In *Proc. MobileHCI* (2014).

[5] Sahami Shirazi, A., Henze, N., Dingler, T., Pielot, M., Weber, D., and Schmidt, A. Large-scale assessment of mobile notifications. In *Proc. CHI* (2014).

[6] Wiese, J., Saponas, T. S., and Brush, A. Phoneprioception: enabling mobile phones to infer where they are kept. In *Proc. CHI* (2013).