Institute for Visualization and Interactive Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Bachelorarbeit Nr. 225

Smart Distribution of Notifications Across Multiple Devices

Philipp Kratzer

Course of Study: Informatik

Examiner: Jun.-Prof. Dr. Niels Henze

Supervisor: Dipl.-Inf. Dominik Weber

Commenced: May 4, 2015

Completed: November 3, 2015

CR-Classification: H.5.2

Abstract

In today's multi-device computing environment a lot of different applications use notifications to inform the user about certain events such as new e-mails, available updates and more. This makes notifications an integral part of interaction with smart devices. However, related work showed that notifications have disruptive effects and are responsible for a large number of interruptions. Thus, with an increasing number of smart devices, including smartphones, tablets, smartwatches and desktop PCs, the negative effects of notifications are exacerbated in the omnipresent multi-device environments of today. As a consequence, there is the need to find a balance between keeping the user informed and confining the negative effects of notifications. For that reason, we developed a framework which is able to select a device or subset of devices to notify the user. Hence, we developed applications which collect contextual information and sensor data from the devices of the user. Furthermore, we conducted an in situ study using the Experience Sampling Method to gain insight into how devices in a multi-device environment are used and to find out which of the contextual information can be used to predict on which device or subset of devices the user would like to receive a notification. Our findings show that there is a strong correlation between the proximity of a device to a participant and whether a notification should be shown on that device. Furthermore, we found out that there are correlations between the collected information and the preferred displaying device for a notification. Based on our findings we implemented a server application for selecting the device or subset of devices as a proof of concept. As a result, the disruptive and interruptive effects caused by notifications in a multi-device computing environment can be minimized.

Kurzfassung

In der heutigen multimedialen Zeit verwenden viele verschiedene Applikationen Benachrichtigungen, um den Benutzer über bestimmte Ereignisse wie neue Emails, verfügbare Aktualisierungen und vieles mehr zu informieren. Dies macht Benachrichtigungen zu einem wesentlichen Bestandteil der Interaktion mit intelligenten Geräten. Existierende Arbeiten zeigten jedoch, dass Benachrichtigungen störend sind und den Benutzer unterbrechen können. Mit einer wachsenden Anzahl an Geräten wie Smartphones, Tablets, Smartwatches und Computern werden die negativen Auswirkungen von Benachrichtigungen noch weiter verstärkt. Folglich muss ein Gleichgewicht zwischen dem Informieren des Benutzers und der Verringerung der negativen Effekte von Benachrichtigungen gefunden werden. In dieser Arbeit wurde ein Framework entwickelt, welches ermöglicht, ein Gerät oder eine Teilmenge der Geräte automatisch zu wählen, auf denen eine Benachrichtigung angezeigt werden soll. Es wurden Anwendungen erstellt, die Daten und Sensorinformationen von den Geräten der Benutzer sammeln können. Um einen Einblick darüber zu bekommen, wie Benutzer mit multiplen Geräten diese verwenden, wurde eine in situ Studie mit Hilfe der Experience Sampling Methode durchgeführt. Ein weiteres Ziel der Studie war es herauszufinden, ob die Daten und Sensorinformationen von den Geräten verwendet werden können, um zu bestimmen, auf welchem Gerät der Benutzer eine Benachrichtigung erhalten möchte. Die Ergebnisse zeigen, dass ein starker Zusammenhang zwischen der Nähe des Benutzers zu einem Gerät und ob der Benutzer auf diesem Gerät benachrichtigt werden möchte, besteht. Des Weiteren wurde festgestellt, dass es Korrelationen zwischen den gesammelten Sensordaten und Informationen und dem bevorzugtem Gerät für eine Benachrichtigung gibt. Basierend auf diesen Ergebnissen wurde eine Server Anwendung entwickelt, die ein Gerät oder eine Teilmenge von Geräten für Benachrichtigungen intelligent auswählen kann. Auf diese Weise lassen sich die störenden Auswirkungen von Benachrichtigungen in Umgebungen mit mehreren Geräten minimieren.

Contents

List of Figures

List of Tables

1. Introduction

Nowadays there is a trend towards owning multiple smart devices such as desktop PCs, smartphones, tablets, smart TVs and smartwatches. Small devices like smartphones and smartwatches are often close to the user and therefore almost permanently available. Also notebooks and PCs are an indispensable part of the daily lives of many people.

A notification is a kind of a message which can be used on smart devices to inform the user about system events, available updates, incoming messages and much more. Therefore a number of different applications make use of notifications on smart devices. Often visual signals, auditive alarms or tactile signals are used to gather the attention of a user.

All of the mentioned smart devices can display notifications and let users interact with them. Thus notifications are a central component and are typically accessible from any screen. Moreover most smart devices have their own, separated notification management. However, in a multi-device environment certain notification types appear on certain devices. For example an e-mail will appear on all devices with an e-mail client in contrast a telephone call would appear on the smartphone but not on the tablet. Hence a user who currently uses the tablet has to check the smartphone to see whether the telephone call is important or not.

Previous work showed that notifications can also have a disruptive effect [IH07, LBGK12, PR15] but users still want to be notified [PR15]. Consequently a possibility to notify users in the least disruptive way would be a desirable goal.

The disruptive effect of notifications in a multi-device environment could be even worse if multiple devices alert at the same time or another device than the one currently used may show a notification, which leads to a higher effort in checking the notification. Therefore to minimize the effect of disruption, notifications should appear on the device or devices best suited for the current context.

In this thesis we design a concept for selecting the most reasonable devices for user notifications based on previously acquired data. Furthermore we want to gain insight into which devices. To this aim we conduct an in situ study using the Experience Sampling Method [CL87]. During the study we periodically ask the participants, with multiple devices, about their current situation and their devices. Moreover we record contextual information and sensor data from these devices. Based on these datasets we implement a system which can select a device or subset of devices for notifying the user, so that the disruptive effect of notifications can be minimized.

Structure

This thesis is structured as follows: In Chapter 2 background information and related work regarding notifications are discussed. Based on this we present our concept of the client-server system to select the device or devices for notifying the user in Chapter 3. In Chapter 4 the implementation of the client applications is described. In Chapter 5 we discuss the design of the study. After that we will discuss the obtained results of the study in Chapter 6. Based on the results, the implementation of the server application is described in Chapter 7. Finally the conclusions based on the results and ideas for future work are presented in Chapter 8.

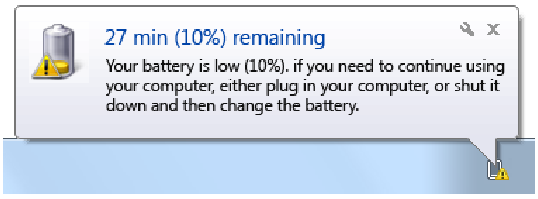
2. Related Work and Background

The following chapter is about background information and related work regarding notifications, contextual information and multi-device environments. First we discuss notifications in common desktop and mobile operating systems and what kind of notifications occur on such systems. We then focus on previous work about disruption and interruptions caused by notifications. After that, we have a look at related work about contextual information regarding smart devices and how the contextual information can already be used to delay notifications. Finally existing multi-device notification frameworks are presented.

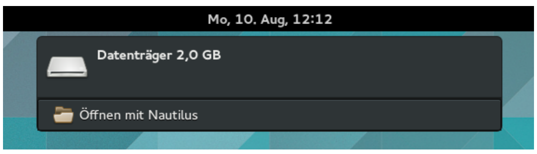
2.1. Notifications on Current Devices

A notification is a kind of a message which informs the user about a certain event such as a new e-mail, available updates and more. Thus a number of different applications use notifications to inform a user. As a consequence notifications are an integral part of interaction on smart devices. Therefore they are implemented in common operating systems for such devices. In this section we will have a look at how notifications on different devices and systems are displayed and how they gain the attention of the user.

Figure 2.1 shows two examples of notifications on common desktop PCs. Figure 2.1 (a) shows a balloon notification from Microsoft Windows 7. It is displayed at the bottom right of the desktop. It consists of a symbol on the left side, a title and a description. Moreover there are buttons in the top right corner of the notification for closing the notification or opening the



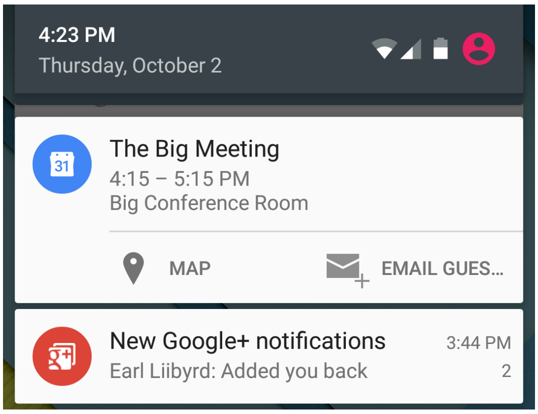
(a) Windows notification[1]



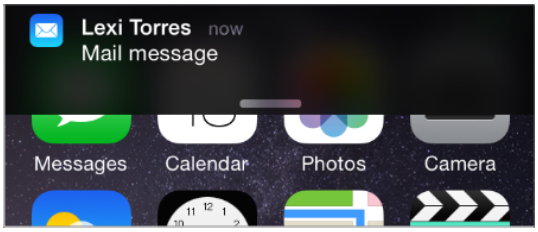
(b) GNOME Shell notification

Figure 2.1.: Examples for notifications on desktop PCs.

[1]Image from: https://msdn.microsoft.com/en-us/library/windows/desktop/ee330740%28v=vs.85%29.aspx



(a) Android notification[2]



(b) iOS notification[3]

Figure 2.2.: Examples for notifications on mobile devices.

notification settings. The notification is only displayed for a few seconds. In Windows 10 there is a feature called Action Center which gives an overview over current notifications and allows to interact with them after they have disappeared.

Figure 2.1 (b) shows a notification from GNOME Shell, displayed at the center top of the desktop. Notifications in GNOME Shell consist of a symbol, the text of the notification and for some kind of notifications there are also buttons at the bottom of the notification. These buttons allow to perform additional actions. For example in Figure 2.1 (b) there is the possibility to us a file manager to open the files of a new attached device. Also in GNOME Shell there is a possibility to view and interact with recent notifications.

Although many desktop environments nowadays have a custom application programming interface (API) for notifications, many applications implement notifications themselves. Therefore a number of different kind of notifications for desktop systems exist.

Figure 2.2 shows two examples for notifications on common mobile operating systems. In Android (a) a notification consists of a notification icon, a title, a text and optional buttons to directly interact with the notification. Since Android 5 they appear on the top of the device and it is possible to open or dismiss them. Moreover there is a notification center accessible by swiping down from the top of the device. Figure 2.2 (b) shows a banner notification from iOS. It also appears at the top and as in Android the user has the possibility to interact or dismiss it.

To gain the attention of the users, notifications have a different behavior depending on the settings of the user and the type of the notification. On desktop systems and mobile devices

[2]Image from: http://developer.android.com/guide/topics/ui/notifiers/notifications.html
[3]Image from: https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/NotificationCenter.html

an alert sound in addition to the visual hint can be played. On mobile devices there is also the option of vibrating. Furthermore different applications can use different alarm settings for their notifications. For example a notification about an update on Android appears silently whereas a phone call may use auditory and tactile signals to attract the user's attention.

Because of the fact that notifications are an easy way to inform users about a variety of events, a number of applications make use of them. Pielot et al. conducted an in situ study about mobile notifications and found out that users receive 63.5 notifications on average per day on their mobile phones [PCO14]. Moreover Ferreira et al. discovered that notifications are a main reason for brief bursts of interaction, called micro-usages [FGK+14]. Sahami et al. conducted a large-scale study of mobile notifications [SSHD+14]. They collected 200 million notifications from more than 40,000 users from an Android application called Desktop Notifications. The application, available in the Google Play Store, forwards notifications from an Android device to a desktop computer. Thus, the notifications pass their server which allows them to collect data and statistics about the notifications. The notifying applications were grouped into categories like messaging, utility, market, etc. Sahami et al. came to the conclusion that users value notifications from messengers and other communication applications more than notifications from system applications. Furthermore users do not like to forward notifications they triggered by themselves. Moreover the click time was recorded and Sahami et al. found out that a notification is clicked on in the first 30 seconds after appearing with a probability of 50% and in the first 5 minutes with a probability of 83%. This indicates that notifications on mobile devices are valued by users. Also Pielot et al. came to the conclusion that notifications were typically viewed within minutes [PCO14].

Another kind of notifications can be seen on some wearable devices such as smartwatches and advanced fitness trackers. They connect to the user's smartphone using Bluetooth and are able to notify the user about new events. Moreover, it is possible to forward all notifications from a smartphone to the smartwatch or dismiss notifications from certain applications. However, smartwatches currently do not have many features besides the possibility to display notifications. Sahami and Henze performed an assessment of notifications on smartwatches [SH15]. They found out that notifications which are considered as most important on smartwatches originate from the calendar and VOIP applications.

Current trends show that the amount of available kinds of smart devices will increase. As a consequence common home appliances become connected which leads to the possibility of having a "smart home". For example several projects developing smart lamps to inform users about notifications are presented on the crowd-funding platform Kickstarter[4]. Smart lamps use so-called ambient notifications, which means that the lamp can for example change its color, to inform the user about an event. Furthermore ambient notifications can be used to slowly make users aware of upcoming tasks while still being able to focus on the current work [MPO13].

[4] https://www.kickstarter.com

2.2. Disruption caused by Notifications

As notifications can appear at any time they can interrupt a user's current work. For example if a user is currently working on some topic and an e-mail notification appears, he or she may switch to his e-mail application to read the e-mail. Iqbal and Horvitz conducted a field study focused on the suspension and resumption of tasks on desktop PC [IH07]. Therefore they logged how users interact with software applications when notifications of the messaging category, high valued by users [SSHD+14], appear. During the study they monitored 27 participants during a two-week period. They found out that users spend on average 10 to 15 minutes responding to notifications. As a conclusion interruptions can delay the completion of a task significantly. Also Cutrell et al. showed disruptive effects of notifications on a variety of computing tasks [CCH01].

Because of the ubiquitous nature of smartphones a notification on such a device can disrupt a user at most of the time. Dey et al. found out that smartphones are within the same room as the user in almost 90% [DWF+11]. In contrast to Iqbal and Horvitz [IH07], Leiva et al. focused on the costs of mobile application interruption [LBGK12]. They conducted a large-scale observation study. The results show that notifications on mobile devices can delay the completion of a task significantly, too.

Pielot and Rello conducted a study of the effect of notifications across devices [PR15]. Therefore they asked 12 people to disable notifications on all of their devices. As a result participants were able to work more concentrated. However, they also felt stressed because of the social pressure to answer messages and they were anxious to miss out important notifications. As a consequence some of the participants also often checked their devices to see if they have received any messages. To summarize, notifications often have a disruptive effect but users still want to get notifications because of social pressure and the fear of missing out something important. Iqbal and Horvitz came to the conclusion that users favor the reception of notifications and therefore accept that they cause disruption [IH10].

To minimize the disruptive effects there are several approaches in related work. Norrie and Murray-Smith conducted a study about the impact of notification displays [NMS15]. They found out that external displays can help making notifications less obtrusive. Moreover notifications can be delayed until an acceptable moment to minimize the disruptive effects [FGB11, HI05, IB08, ORN+15, SD14]. Also subtle notification cues can be used to inform an user less-intrusive [HL00] or an enhanced interaction design can lower the impact of phone calls [BLG+14]. Sharing contextual information makes it possible to avoid at least some disruptive notifications such as phone calls at all. Knittel et al. conducted a survey about sharing contextual information between a caller and a callee [KSSHS13] and came to the conclusion that people are willing to share some contextual information.

The effect of disruption may additionally get worse with an increasing amount of smart devices. More devices which alert the user makes it more difficult to ignore a notification. Moreover in a multi-device environment more than one device can notify about the same event. As the alerts may be delayed, this leads to unnecessary interruptions and exacerbates stress and disruption. Also the need to switch the device for checking a notification may occur more often which results in additional costs. Furthermore a higher amount of smart devices leads to more notification sources. As a result more notifications appear and thus the user may be interrupted more often.

2.3. Contextual Information and Devices

In mobile devices such as smartphones a number of sensors for example acceleration, location or proximity sensors are usually included. This makes it possible to obtain information of the device's context. For example Wiese et al. explored where people keep their phones and made use of low-cost sensors to classify these places [WSB13]. For example, places they looked at are pocket, bag, car, hand, etc. They came to the conclusion that reasonably accurate classifications are possible with common sensors like accelerometers and proximity/light sensors. Such information can be used to show a notification on the device when the user holds it and send it to another device if it is on the table.

Another task to use context information regarding notifications is to find a good moment for a notification. This means that notifications are stored until an acceptable point in time for notifying the user is reached. Bailey and Konstan found out that delaying the delivery of notifications until an acceptable point is reached can decrease the disruptive effect of notifications [BK06]. Ho and Intille developed a mobile computing device that automatically detects postural and ambulatory activity transitions [HI05]. To find such transitions an external acceleration sensor was used. They found out that messages which are delivered at such transitions are received more positively than the same messages delivered at random times.

Another concept of scheduled notification management was developed by Iqbal and Bailey [IB08]. They defined several breakpoints such as saving or switching a task. Iqbal and Bailey found out that if notifications are delivered at such a breakpoint, there is less reaction time. Thus, if notifications only are shown when such a breakpoint is reached, it may result in less frustration of the user.

Moreover Smith and Dulay used machine learning techniques to delay a disruptive smartphone notification or change the intensity or mode of an alert [SD14]. Okoshi et al. looked at a multi-device breakpoint detection for smartphones and smartwatches [ORN$^+$15]. The breakpoint detection is realized with UI-based and activity-based breakpoint detection techniques.

2.4. Multi-Device Notification Frameworks

While there is a substantial corpus of work regarding notifications on single devices, previous work on the behavior of notifications in multi-device environments is limited.

Horvitz et al. already mentioned a notification platform in 2003 [HKPH03]. The platform is able to distribute messages from multiple message sources over multiple devices such as PDAs and PCs. In 2008 Arlein et al. proposed an adaptive notification framework architecture which allows to distribute notifications across multiple devices [ABBE08]. This architecture was prototypically implemented for personalized notifications over a public network as well as over a private network in an e-Health domain.

Weber et al. developed a multi-device notification system which is able to share notifications across multiple devices such as smartphones, tablets, PCs and smart TVs in 2015 [WSH15]. This system is used to gain a deeper understanding of notifications. The authors report that the system distributes notifications by almost 30,000 users per day. The system broadcasts notifications from the users' smartphones and tablets to other devices like smart TVs and desktop PCs. The researchers use the system to collect data from multiple devices. However, broadcasting notifications can make the effect of disruption even worse because more devices alert the user at the same time. It may be better to make use of contextual information to systematically select devices to distribute notifications to.

2.5. Summary and Discussion

In this chapter we looked at related work regarding notifications, disruption of notifications and contextual information of devices. First we discussed the implementation of notifications on common operating systems for smart devices. Studies have shown that notifications are an often used and important part of interaction on smart devices [PR15, SSHD+14]. However, they also have a disruptive effect [IH07, LBGK12, PR15]. In multi-device environments these disruptive effects may additionally be increased. Although there are negative effects, users still wish to be informed because of social pressure and the fear of missing out something important [PR15].

To minimize the effect of disruption there is the need to find the right balance between keeping the user informed and limit the negative effects. On the one hand the effect of disruption may increase with the amount of devices alerting the user about an incoming notification. That is the reason why current multi-device notification frameworks [HKPH03, WSH15], which alert all devices at the same time, may not be ideal. On the other hand the user is not able to receive a notification as soon as possible if it is sent to the wrong device.

Therefore we want to explore if it is possible to automatically determine the most suitable subset of devices for a given context so that ideally only the best suited device alerts the user when a notification is incoming. Thus we decide to develop a model for selecting the device or devices on which the user most likely wants to receive the notification on.

Related work showed that contextual information can be used to find out where a smartphone is [WSB13] or when a good point in time for a notification is reached [HI05, IB08]. Hence, we want to select the device or devices for the notification, depending on the contextual situation of the user and the contextual situation of the devices. We try to obtain these contextual information using data collected on common smart devices from user input, system events and sensor data. Moreover we want to gain a deeper insight into notifications in a multi-device environment in connection with contextual information.

3. Concept of the System

In the previous chapter we came to the conclusion that notifications have a disruptive character. Therefore we want to confine these negative effects by selecting only one device or a small subset of devices to alert an user about a new notification. Therefore, we require the possibility to have a decision making algorithm which considers contextual data from all the devices at a centralized point. Thus in this section, we will have a look on the concept of a prototypical system for selecting the device or devices and for transferring contextual data. First we discuss the general idea of the system. Afterwards we look at how the devices for notifying the user can be selected. Finally we have a look at the overview of the prototype.

3.1. Concept of the System

In this thesis we focus on users with multiple smart devices. The most common smart devices currently are desktop PCs/notebooks, smartphones and tablets. Also the amount of smart TVs and smartwatches is increasing. We focus on PCs/notebooks, smartphones, tablets and smartwatches because they innately are capable to display notifications.

If a notification raises at a centralized point, it should be forwarded to the user's device most suitable for the current situation. For example a notification can therefore be forwarded to the device actually running. However, if currently more than one device is running, it could be forwarded to the device currently active. An active device can be determined using information about the context of the device. We want to collect such information using sensors and other available data from the devices. For example an active device can have the display enabled or the user may currently interact with the device. However, it is unclear how the device should be selected if more than one device is active or none of the devices is active. Thus we will have a detailed look on the way how to select the device for a notification in the following section.

3.2. Selecting the Device(s) for a Notification

A central question of this thesis is how the device or devices for notifying the user are selected. Thus we make the assumption that the choice of the device or devices depends on the current

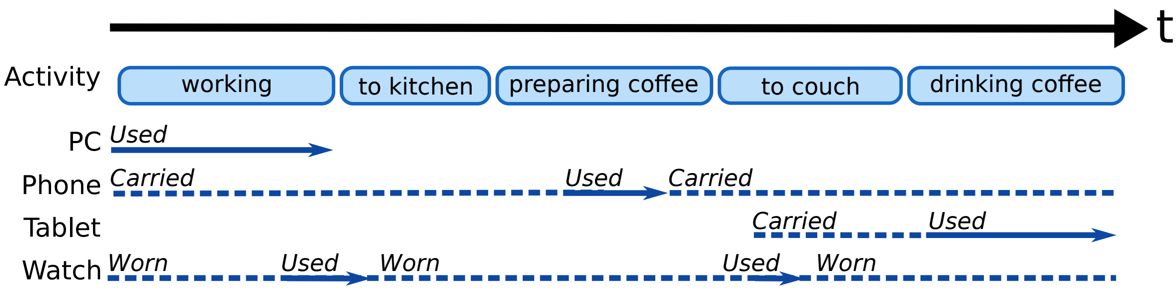


Figure 3.1.: An example scenario how devices could be used over time.

context of the user and his devices. A consequence of this assumption for making a good selection of the devices is that information related to the following questions is required:

- What is the user doing and where is he? (user's context)
- Where are the devices and in which state? (devices' context)

Because users like to receive notifications [PR15] a possible criterion to select the device or devices is the proximity of the user to a device. Thus the cost of switching between devices is reduced. Moreover a notification will reach the user more likely if the device it is sent to is nearby the user. To use the proximity criterion detailed information about the position of the user and the devices is required. Moreover it is not clear which device should be notified if there are more devices with the same proximity.

Figure 3.1 shows a scenario of how the devices of a user could be used over time. First the user is working at the PC. At this time the PC may be a good choice for notifying the user because no device switching is required. After that the user looks at the smartwatch and walks to the kitchen. During that time the smartphone or smartwatch should be preferred over the PC. While preparing coffee the smartphone is used and after that the user walks to the couch for drinking coffee and using the tablet. As a consequence the tablet can be seen as a good device for a notification because it is in proximity to the user.

To find out whether the proximity criterion holds and if it holds, how it could be improved and implemented, we conduct a study using the Experience Sampling Method (ESM) (see Chapter 5). With this study we want to obtain a deeper understanding of the user's demand for notifications in multi-device environments. Moreover we want to find out which information is suitable for an algorithm which selects a device or devices to notify the user. Thus we applications for collecting data of the user's and the devices' context is required.

We want to obtain this contextual data from the devices. Therefore sensors and system events from the different smart devices have to be collected. For example system events such as *log in* or *log out* can mean that the user begins or ends to use a device. The acceleration sensor for example can be used to see whether a device is currently on the move. If certain devices are

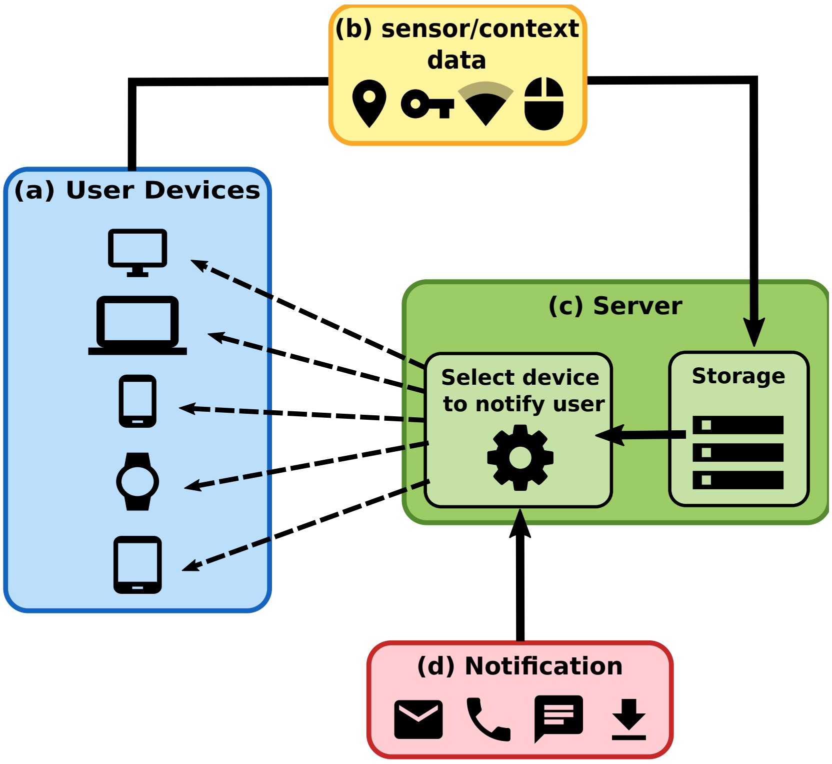


Figure 3.2.: User devices collect sensor and context data and send it to the server. The server evaluates the data to select the device or devices on which the notification should be displayed.

on the move while other devices are motionless, it is likely that the devices in motion are with the user and a notification may only be recognized if it is sent to that device.

3.3. Overview of the Framework

The contextual data collected on the smart devices of a user can be used for determining the device for notifying the user. Thus the data from the different devices must be combined and analyzed. As a consequence we need to store the data on a central server which has the possibility to run an algorithm for decision making. Therefore it is required to design a framework which is capable of multi devices and has the possibility to select the device or devices for notifying the user.

Figure 3.2 shows a conceptual overview of the system for a single user. To obtain information about the current context of the user's devices, an application collecting sensor data and contextual information has to run on each device (see Figure 3.2 (a)). In addition, collected data is sent to a server (see Figure 3.2 (b)). The data from the devices is stored at this central point (see Figure 3.2 (c)) which leads to the ability to use the data for decision making. If a notification (see Figure 3.2 (d)) is incoming, the server decides to which device or devices it should be forwarded.

Because the focus is on the system needed for selecting the device or devices to notify the user, it is not further specified where the notification is coming from. A conceivable possibility is to adapt an existing multi-device notification framework (e.g. [WSH15]) to obtain the notifications from the user devices themselves. Another opportunity is to send notifications from external sources directly to the server or let the server pull notifications from external sources (e.g. e-mail). The overall goal is to have the server as a powerful notification processor which allows to receive notifications from various sources and functions as a notification managing instance. Such a holistic notification system is capable to distribute the notifications in a intelligent way across devices.

3.4. Summary

In this chapter we introduced a proposal for a system to select the device or devices for notifying the user. On the devices of the user, applications collecting information about the devices' context and the user's context, are installed. The information is obtained using sensor data and system events. Furthermore the data is transferred to a central server which uses the information for decision making. If a notification is coming in, the data is combined and the best target or targets to forward the notification is determined. In the following chapter we will have a look at the implementation of the applications for collecting the data.

4. Implementation of the Android and Windows Applications

To make considering contextual data and sensor information possible it is required that applications on the client side forward such data. In this section we describe the implementation of the client-side of the system. The clients collect useful contextual information and send it to the server. We will first discuss possible sources for contextual information and their usefulness. After that we will have a look at the implementation of the Android application and finally on the Microsoft Windows application.

4.1. Sources for Contextual Information

There are various possibilities to obtain contextual information. However, we focus on existing and widely used devices because additional hardware leads to an additional burden, if users would first have to buy and set up the hardware. Moreover smart devices nowadays are equipped with sensors making it possible to obtain detailed information about a device's context. Thus in a multi-device environment data from each device of a user can be obtained. However, it is more demanding to get data about the context of the user him- or herself because users may leave the sensor range of their devices.

Depending on the device there is different useful data making it possible to draw conclusions about the context of the device. Table 4.1 shows an overview of different data sources. Not all of them are available on all devices and their serviceability depends on the device. For example on the smartphone, the *screen on* event is most of the time triggered through the user by using a hardware button, while on the smartwatch a movement of the arm may cause the event unintentionally.

The first source of contextual information we look at is information obtained from user input (see Table 4.1 (a)). This data may be useful because the device has to be actively used to produce direct input. Thus this section provides information about both: the device and the user. If the user is currently using a device, he or she could be addressed on that device. Therefore it might be a good choice to notify the user on that device. Moreover it is more likely to reach a user on a device recently used than on a device which was not used for hours. Thus the possibility to see which device was used last also leads to suitable information. However,

(a)	User Input	Touch Input
		Mouse Movement
		Keyboard Input
		Speech Recognition
(b)	System events	Power on/off
		Login/Logout
		Wi-Fi/Mobile connection
		Headphones
		Charging
		Screensaver becomes active(inactive)
		Display on/off
		Foreground application
(c)	Sensor data	Acceleration Sensor
		Proximity/Light Sensor
		Microphone/Camera/Webcam
		Position Sensor
(d)	Application data	Calendar Information
		User's Messages/E-mails

Table 4.1.: Overview of data sources about devices' and user's context.

there is the opportunity that the user leaves the device and does not return immediately. Moreover the user might be watching a video and is not longer interacting with the device. As a conclusion it is not sufficient to only consider data from the user input section.

Another data source which can be considered are system events (see Table 4.1 (b)). These events are in most cases consequences of actions or activities of the user. In contrast to the user input section (Table 4.1 (a)), these events do not require the user actively using the device. For example the user can press the power button and after the process of booting is finished, the start event will raise. It is likely that the user is going to use this device but it is also possible that he or she just pressed the button and walked away. For the *Wi-Fi connected* event, the user does not need to press a button of the device to raise it because it is enough to carry the device into the Wi-Fi range. It is also possible to trigger the event just by activating the Wi-Fi access point. Data obtained from system events provides more information about the device than the user. For example, one is able to detect that the device is at a certain area (area of the Wi-Fi access point) but one doesn't know whether the user is still in that area or has already left it.

As already mentioned, a number of mobile devices especially smartphones, smartwatches and tablets have integrated sensors (see Table 4.1 (c)). The purpose of these sensors is to detect events or changes in the environment. Thus data recorded from sensors can be used to obtain information about the device's environment. For example acceleration data can be used to find out whether the device is currently in motion or not. Moreover, the camera can be used to detect whether the user is currently nearby the device using face recognition algorithms. However, some of the datasets are to be considered critically because of the user's privacy. Users rarely like to tell their exact position or want the camera or microphone to record constantly. Because mobile devices are limited by their battery capacity, another challenge is the high power consumption required to record sensor data. As a consequence sensor data can not be recorded permanently. Furthermore not all sensors are available on all devices.

The last section of the table (see Table 4.1 (d)) is related to data obtained from applications and custom input a user has entered such as calendar entries or messages. These inputs can be parsed to extract information about the user's context. For example a calendar entry at a certain location implies that the user is at the particular location at the corresponding time. However, there is usually no information about the devices the user will use at that location. Moreover a calendar may only include major appointments and there is the possibility that some appointments are wrong because of spontaneous changes. Furthermore some users may not have a digital calendar or use the calendar just rarely. Another problem in this section are privacy issues, too, because users may not allow their messages or calendar to be parsed [FEW12].

As a result information from the mentioned categories can be suitable to predict the device or subset of devices the user wants to be notified on. Therefore the client-side applications running on the user's devices should try to obtain detailed information from these categories. However, information obtained from the mentioned sources is more related to the devices' context than to the user's context. The user's context often has to be guessed on the basis of the devices' data, if needed.

4.2. Android Application

We decided to use an Android application for smartphone, tablet and smartwatch because Android is a widely-used operating system for the mentioned types of devices. The task of the application is to collect as much useful contextual information as possible and send it to the server.

The main part of the Android application is a service. A service is an application component running in the background and does not provide a user interface. However, the application also consists of a main activity, which represents a user interface, but is primarily used for debugging purposes. So that users can see when contextual information and sensor data are recorded, the application registers the service as a foreground service. Thus an icon appears at the top of the screen when the service is running. Moreover a foreground service gets a higher priority from the operating system and therefore has a longer lifetime.

To get contextual information via the Android application it is not possible to collect data from the sensors all the time because it would drain the battery rapidly which is bad because mobile devices are limited by their battery life. Therefore we decided to focus on data from the system events category (see Table 4.1 (b)). Thus, turning the display on and off, change of the connection status, power on/off and plug in/out of headset and power cable are recorded. These events are triggered by the system and therefore no permanent “listening” is required.

Moreover we recorded event-based sensor information using the location application programming interface (API) from Google Play Services [goo]. These APIs are optimized for battery life and allow to register a timer to periodically get certain sensor information. We used it to obtain location and activity recognition data. The activity recognition data consists of a list of activities such as *in vehicle, on foot, bicycle, still, running, tilting, unknown, walking* that a user may have been doing at a particular time. A confidence for each detected activity indicates how likely the activity is.

Direct user interaction with an Android device is primarily done via touch interaction. Since the Android API for security reasons does not allow to register for system wide touch events, we put an invisible window above the application area. With this workaround it is possible to determine whether the user is interacting with applications. However, if he or she touches the lock screen or the system buttons, no touch event is triggered because the invisible window does not overlay system components.

The data is sent to the server as soon as new data appears. If it fails or there is currently no Internet connection, it is written to a text file. Thus it can be retried to send it later without a loss of data. To identify at which time a certain event was raised, a timestamp is added to all data. Because the sending process is a source of errors and we want to be able to collect data even without Internet connection, we included an optional possibility to always write the data to a text file instead of sending it to the server for the study (see Chapter 5).

To offer an easy way to install the application, it was published in the Google Play Store. Because the Android application should run on a wearable too, an extra module for the smartwatch was included. Thus when installed on a smartphone, the application is automatically installed on Android Wear based smartwatches. However, the module of the smartwatch is almost the same as the original application. Only a possibility to transfer the data file from the smartwatch to the smartphone via Bluetooth was included.

4.3. Windows Application

For desktop PCs we decided to develop a Windows application because Windows is the most used operating system for that type of device. The Windows application is written in C# and consists of a single background process. Similar to the Android application, it has a permanent tray icon in the status bar while running.

In contrast to smartwatch, smartphone and tablet, less sensors are included in common desktop PCs or notebooks. Therefore we do not collect sensor information with the Windows application. Moreover from the system events only the categories *login* and *logout* are captured. Such as the Android application, the Windows application has the possibility to send the data directly to the server or write it to a text file on the desktop.

The user interaction is recorded with a screensaver like approach. The application logs when the user is starting to interact with the device. Moreover it is logged when there was no interaction for at least one minute. Therefore it is possible to find out afterwards when the user was interacting with the device. As an addition we also log the name of the current foreground program to when the user is inactive because there is the possibility that the user is not interacting with the device but still sitting in front of it. This is the case, for example, when the user is watching a video. Then a video player would be in the foreground.

The application creates an entry in the startup section of the Windows registry to make it possible to automatically start the program when Microsoft Windows has booted. To provide an easy-to-use installation of the application, the *ClickOnce* deployment technology integrated in *Microsoft Visual Studio* was used. Using that technology no administrative permissions are needed to install the application. Moreover *ClickOnce* takes care of external dependencies such as the .NET framework and asks for installing them if required.

4.4. Summary

We discussed different sources for obtaining contextual information. The different data sources are not all available on all devices and they are different useful depending on the device. Moreover we looked at the implementation of the Android application for smartphones, smartwatches and tablets and the Windows application for desktop PCs. The applications record sensor data and contextual information from the devices and can send them to a server or write to a log file. Thus using the applications detailed in this section, it is possible to conduct a user study. The obtained data can lead to valuable information about the user behavior in multi-device environments. Moreover we want to use the data to find a possibility to predict on which device the user most likely wants to receive a notification.

5. Design of the Study

To get a deeper insight into the user behavior with multiple smart devices, we conduct an in situ study using the Experience Sampling Method (ESM). Moreover we want to find out what kind of contextual information can be used to determine the best device or subset of devices to notify the user. In this chapter we first describe the design of the study. After that we discuss the survey application the study participants answer during the day.

5.1. Setup

For the study we decided to use four different smart devices per user: an Android smartphone, an Android tablet, a Windows desktop PC or laptop and an Android Wear-based smartwatch. We chose smartphone tablet and PC because these devices are widely distributed. Smartwatches are not that wide-spread yet but with an increasing amount of available devices, more and more users are adopting this new technology.

Because we want to get the most natural behavior, the participants should use their own devices. However, of the recruited participants only one was owning a smartwatch. Therefore we decided to give each of the participants a smartwatch. A smartwatch is learnable very fast and thus does not have a long acclimatization period. Because some study participants did not own a tablet, they were equipped with one.

On all the devices our data collecting applications were installed. They were adjusted to write everything in a text file instead of sending the data to a server to avoid the need of an Internet connection during the study. Moreover buttons for exporting and sending the log files via e-mail were added. On a device with a smartwatch connected, the data file from the smartwatch is first transferred to the handheld and following both files can be sent via e-mail. All study participants were advised to use the devices we gave them as if they are their own. Furthermore, they were told to use all the devices like they would use them when not participating in the study.

For the study we decided to use the Experience Sampling Method (ESM) which is an research methodology that interrupts a participant at certain times and asks them to fill out brief questionnaires. This has the advantage that no additional cognitive load for the participants is required and that they will act in a natural way during the time of the study because they do not know when they have to answer a questionnaire. Thus, we need to interrupt the study participants in their practices of daily life and ask them periodically about their context and on which device they want to get notified. As a consequence an opportunity to show them a survey is required. Because there are questions about the devices in the survey, opening the survey on one of the devices of the participant would falsify the results. Moreover opening the survey on all devices at once needs synchronization of the devices which is not possible if one of the devices currently does not have an Internet connection. So we decided to hand out an additional device for the survey to the study participants. To save battery power the additional device was set to airplane mode and all background services, which were not required for the survey application, were disabled.

For the study, we scheduled a period of seven days because we wanted to capture each weekday completely due to the fact that the behavior of the participants may change depending on the day. For example on weekends the daily routine might differ compared to workdays. The setup of the study and the handing back of the devices was carried out one day before and after the seven days, so we were able to collect seven full days. Furthermore, setting up the devices one day before the actual data collection allowed the participants to familiarize themselves to the survey device, samrtwatches and in some cases tablets.

5.2. Experience Sampling Survey

The survey consists of four questions as shown in Figure 5.1. Because all the participants were German, a German translation of the survey was used. The first two questions are about the participant's surroundings. Question one asks the participants where they are. The possible answers are *in transit*, *at home*, *work/uni*, *cafeteria/cafe/restaurant*, *sport* or *somewhere else*. If the option somewhere else is selected an input box to add, additional information appears. It is also possible to select more than one answer at the same time because it is possible that more than one option applies. For example somebody can work at home and is traveling on business.

The second question asks the participants how many persons are in their proximity. The answers are graded into categories. Possible answers are 0, 1-3, 4-10, 11-50 and more than 50.

The third question is about the proximity of the participant's devices. For each device there is the possibility to rate the device using a 5-point Likert scale. Thus the participant has to answer for each device with a value between 1 and 5 where 1 is "I do not agree" and 5 is "I strongly agree". Furthermore we assigned an example to each point to make answers from different persons better comparable. Thus a legend under the question was added. It assigns 1 to "currently unreachable", 2 to "walking and higher effort required", 3 to "walking required", 4 to "movement required" and 5 to "reachable".

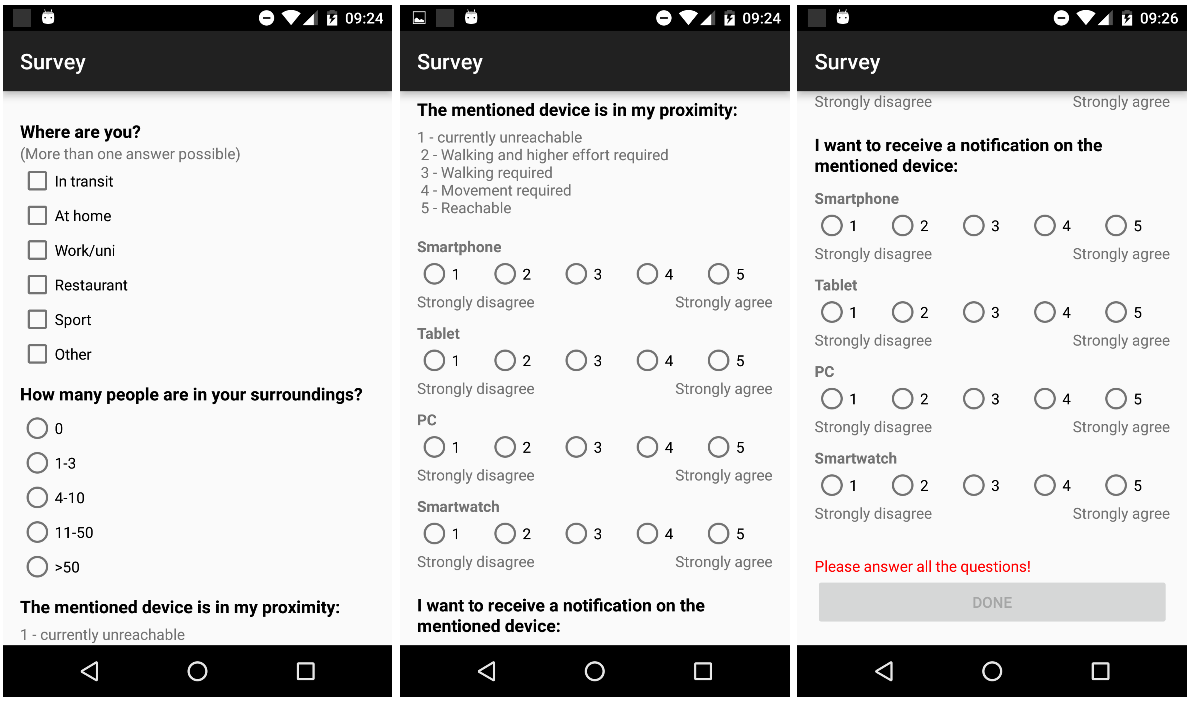


Figure 5.1.: The survey periodically opening on an additional device. It consists of four questions about the context and devices.

The last question is to find out on which device the study participant want to be notified. Therefore the question asks the participants on which of the mentioned devices they want to get a notification. In this question the study participant has for each device the possibility to answer with a 5-point Likert scale between 1 (disagree) and 5 (strongly agree). Thus the participant should put in relation how likely he or she want to get a notification on one of the devices.

The survey periodically opens randomly every 45 to 90 minutes. It shows an icon in the notification bar on the top left. With clicking on it the survey activity opens. It uses the standard alert sound of the device to gain the participants attention. Thus, the participant can set the device to vibration or silent if the standard sound is disturbing or not appropriate. Moreover no survey appears at night between 0am and 6am. The survey notification stays available for 10 minutes before disappearing again. The results are only stored if all questions are answered. At the end of the study each participant received a monetary reward of EUR 0.20 for each completed survey. All the answered questions are stored with a timestamp in a text file.

5.3. Summary

In this chapter we looked at the design of the user study. For the study a period of seven complete days was scheduled. With the study we like to provide an overview of notifications in a multi-device environment. Thus the participants of the study were equipped with smartphone, smartwatch, tablet and PC. They used their own devices if they had one, otherwise they were provided with the particular devices. We used the Experience Sampling Method and opened a survey at random times. The survey consists of questions about the participants' surroundings, the proximity of their devices and on which device they want to receive notifications. Moreover the applications, introduced in the previous chapter, were installed on the devices of the participants. Thus we are able to obtain contextual information and get a deeper insight into the usage of the devices. In the following chapter we will look at the results of the study.

6. Results of the Study

In this chapter we discuss the results of the study. First we have a look at the participants and how they used the devices. Afterwards we focus on how many and when the surveys are completed. Then we discuss the answers of the survey regarding the surroundings of a participant. After that we have a look at the proximity of the devices and afterwards on which device the notification should be shown on. Last but not least correlations between the questions of the survey will be analyzed. Finally we discuss whether the contextual information obtained from the devices can be used to decide where the notification should be sent to.

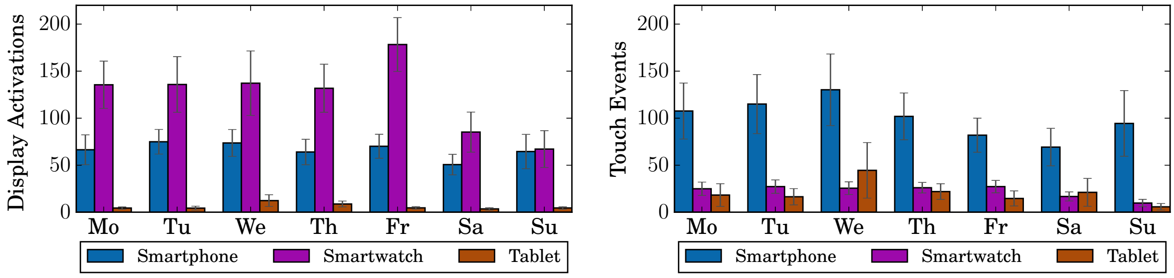
6.1. Participants

The study participants were recruited from the university campus and the university mailing list. We found twelve study participants but we only used results of participants from whom we were able to retrieve a complete set of log files and who completed more than ten surveys. Therefore we rejected two people. In one case only one survey was answered and in the other case we were unable to receive the log file from the smartwatch. As a consequence ten study participants (two female, eight male) aged between 21 and 27 ($M = 23.6$, $SD = 2.32$) contributed to the study. Seven participants were students. We looked for participants who own an Android smartphone, an Android tablet and a Windows PC. Due to the lack of volunteers that own a tablet, two participants did not use their own tablet during the study. Moreover two people participating at the same time shared their tablet and therefore used two different accounts on that tablet.

6.2. Device Usage

During the study the participants had four different kinds of smart devices. Because these devices vary in size, weight and mobility, also the way of interaction with this devices vary.

Figure 6.1 shows how the Android based devices were used on average. Figure 6.1 (a) shows how often the display of a device was activated during the day. Most often the display of the smartwatch has been activated ($M = 124.41$, $SD = 87.79$). A possible reason for this phenomena is that the display of the smartwatch switches on if the arm of the participant



(a) Display switched on

(b) *Touch* events

Figure 6.1.: Shows the number of *display on* events and *touch* events occurred on average with standard error for each day of the week and for all participants.

does a certain arm-swing. Thus the event triggers more often than on devices that require an explicit button press. Moreover the smartwatch is worn at the arm and therefore it is more easy to check the time or notifications by looking at it instead of using the smartphone or the tablet. It can be seen that the smartwatch was activated more often during the week than during the weekend. The high value of the standard error is caused by differences between the participants in using the device and by the fact that sometimes study participants did not wear the smartwatch for a full day. The smartphone display was activated the second most time ($M = 66.3$, $SD = 43.74$) and the tablet display was activated least often ($M = 6.04$, $SD = 9.38$). There are no values for the PC because no *display on* events and *touch* events were recorded.

The occurrences of the *touch* event on a certain day can be seen in Figure 6.1 (b). A *touch* event is raised when the participant touches the display and no *touch* event has been raised for at least one minute. In contrast to the *display on* event there were more *touch* events on the smartphone ($M = 100.03$, $SD = 89.73$) than on the smartwatch ($M = 22.46$, $SD = 19.44$). This may be a result of the fact that it is more comfortable to write messages or browse the web on the smartphone because of the bigger screen size and the ability to use both hands for interaction and text input. Also one study participant mentioned that he did not like the smartwatch for answering text messages and therefore just used it to check if a message is available and then switched to the smartphone to compose an answer. Another study participant said that he missed a way to write a message to somebody using *WhatsApp* without them writing him previously.

Moreover the smartphone registered more touch events per day than the tablet ($M = 20.34$, $SD = 45.03$). A reason for this is that the smartphone is, because of it's size, more often with the user and is therefore used more often.

Figure 6.2 shows the average duration of user interaction with the four devices per day. For the PC the time the participant performs direct input was recorded. The direct input starts if the participant moves the mouse or presses a key and it stops if the participant hasn't moved

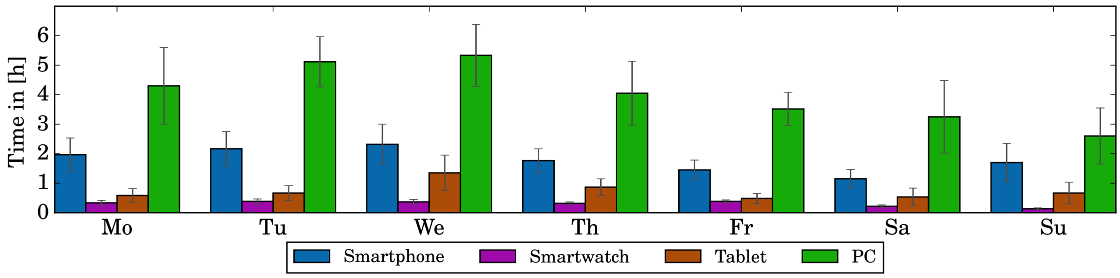


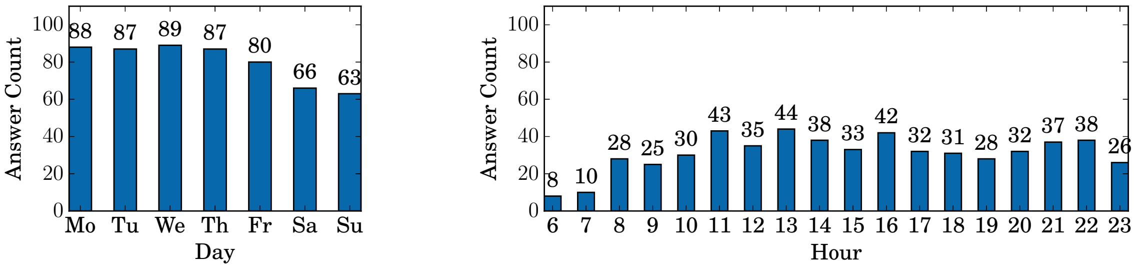
Figure 6.2.: The average time the participants used the devices are shown for each day. On the Android devices (smartphone, tablet and smartwatch) the total time when the display was activated was measured, for the PC the time the participants interacted with the PC was recorded.

the mouse or pressed a key for at least one minute. Because smartphones and smartwatches are often just used without user input, for example for checking the time or looking for new messages, the same method for measuring activity is not sufficient. Therefore we recorded the time the display was activated. As a consequence the duration measured for the PC can be seen as a lower limit because there is the possibility to use the device without user input, for example while watching a video. In contrast the duration measured for the Android devices can be seen as an upper limit because if the display is turned off, there is no interaction with the devices.

As seen in Figure 6.2 the PC was the longest used device on each day ($M = 4.01h$, $SD = 3.16h$), although just the pure interaction time was measured. The next device is the smartphone with an average screen on time of $1.47h$ ($SD = 1.39h$) per day, followed by the tablet ($M = 0.44h$, $SD = 1.04h$) and the smartwatch ($M = 0.18h$, $SD = 0.14h$). The high usage of the PC is due the fact that most participants were students or employees who learned or worked with the PC.

Moreover it can be seen that the devices on weekdays from Monday to Friday on average were more used than on the weekend. For example the smartphone was used on average on weekdays for $1.56h$ ($SD = 0.08h$) and on the weekend it was used $1.26h$ ($SD = 0.14$), the smartwatch was used for $0.21h$ ($SD = 0.01$) on weekdays and for $0.10h$ ($SD = 0.01h$) on the weekend, the tablet for $0.47h$ ($SD = 0.10$) on weekdays and $0.36h$ (0.02) on the weekend and the PC was used on weekdays for $4.28h$ ($SD = 0.16h$) while it only was used on average for $2.55h$ ($SD = 0.12$) on the weekend.

The average interaction duration of the smartwatch was $0.09min$ ($SD = 0.02min$), of the smartphone $1.24min$ ($SD = 0.42min$), of the tablet $8.29min$ ($SD = 9.08$) and of the PC $9.36min$ ($SD = 4.44min$). The values of the Android devices were computed using the time from enabling the display until disabling the display, for the PC the time from the start of user input until the end of user input was used. It can be seen that on average the smartwatch and



(a) Completed surveys per day

(b) Completed surveys per hour

Figure 6.3.: Sum of completed surveys per hour and day from all participants.

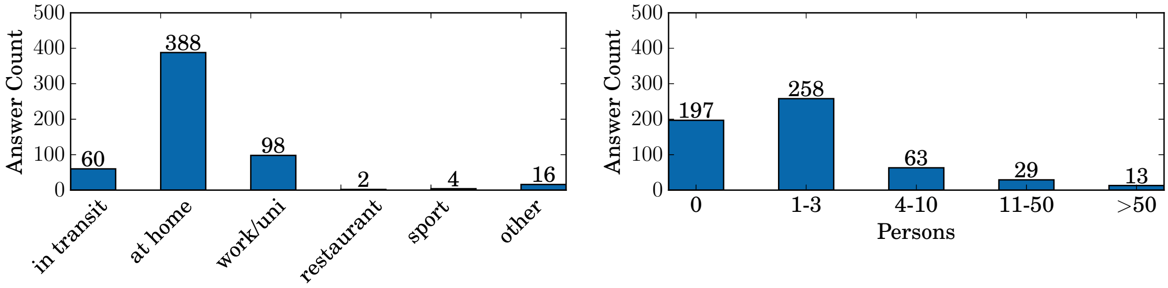
the smartphone are used for very short interactions while the tablet and the PC are used for longer interaction periods. Also previous work found out that the smartphone is very often used for short interactions [FMK+10], and that these so called micro-usages are dependent on the user's context [FGK+14].

To summarize, the smartwatch is the device with the most *display on* events but with the smallest total display activation time. In contrast the smartphone and the tablet have less *display on* events but a longer display activation time than the smartwatch. A reason of that is the undesired activation and that the smartwatch is used for short but often interactions like checking the time. The smartphone is also used for short sessions but has in total more time with enabled display than the smartwatch and the tablet and it also has the most *touch* events. A reason for this is the form factor, as the smartphone is typically with the user and is more comfortable to write messages, browse the Internet or play a game. In contrast tablet and PC are used for longer sessions. However, the tablet has the fewest *touch* events and the fewest *display on* events. As a consequence it is the least used device. In contrast the PC is the device the longest used.

6.3. Completed Surveys

The participant completed between 14 and 82 surveys. In total 560 surveys were completed ($M = 56$, $SD = 20.09$). The study participants sometimes missed surveys because the survey device was not in their proximity, it was silenced or their current situation did not allow to answer a survey. Therefore the count of completed surveys varies depending on the day and the hour of the day.

Figure 6.3 (a) shows the total number of surveys completed at a certain day. On average 80 surveys were completed at a day with a standard deviation of 11.06. At weekdays from Monday to Friday slightly more surveys ($M = 86.2$, $SD = 3.56$) were completed than on the weekend ($M = 64.5$, $SD = 2.12$). Figure 6.3 (b) shows the total number of surveys completed



(a) Location

(b) Persons in the surroundings

Figure 6.4.: An overview of the answers of the first two questions of the survey about the location and the persons in proximity to the study participant. The values are the total count of surveys which were answered with the corresponding category.

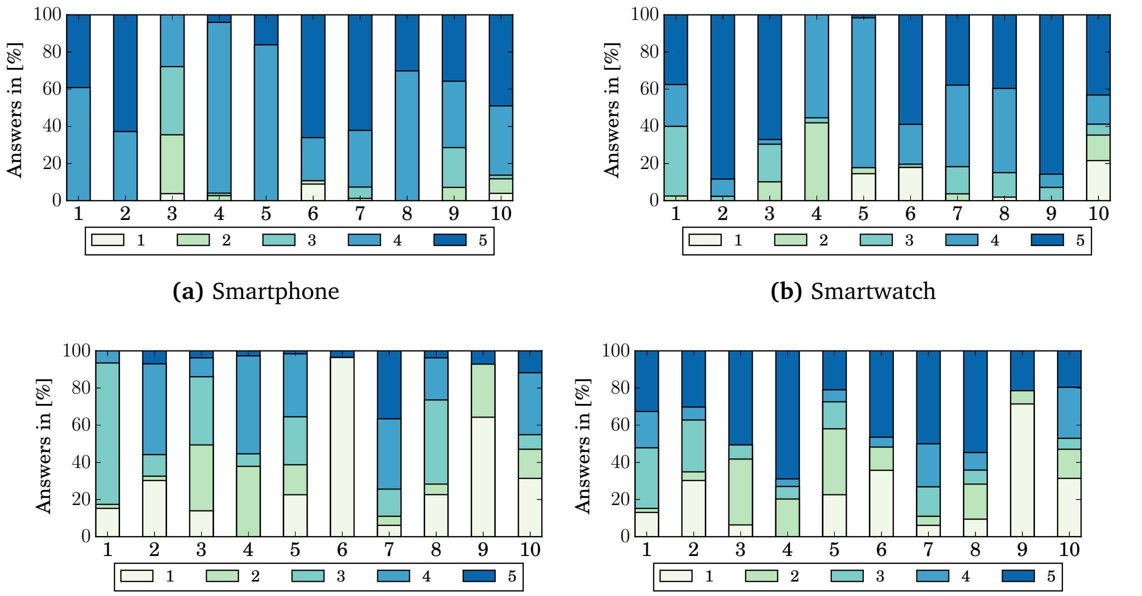
at a certain hour. There are no answers between 0am and 6am because no survey was triggered at this time. It can be seen that few surveys were completed in the early morning (6am-8am) and most surveys were completed about noon (11am-14am).

6.4. Participants' Surroundings

Because the study participants were interrupted at a random point of time in their daily life, they were on different locations with an varying amount of people in their surroundings while answering the questions. To gain an insight about the participants' surroundings, the first two questions of the survey were about where the participants currently are and how many persons are in their proximity.

Figure 6.4 (a) shows the results of the questions about the location of the study participants. It was possible to select more than one option for this question, therefore the sum of answers is higher than the number of completed surveys. It can be seen that most of the surveys were completed *at home* (68.31%), followed by *work/uni* (17.25%) and *in transit* (10.56%). This can be a result of the participants being most of the time at home or having less possibilities to answer a survey at the other options . For example one study participant stated that he was not able to complete a survey during a meeting at work. Another participant said that she was not carrying the survey device during sports and therefore may missed surveys at that time.

Figure 6.4 (b) shows the result of the question about the number of persons in proximity to the participants. In contrast to (a) it was not possible to answer more than one category. Most surveys were completed with 1-3 persons in proximity (46.07%) followed by the option with 0 persons in proximity (35.18%) and 4-10 persons in proximity (11.25%). It can be seen that most surveys were answered with less than 4 persons in proximity (81.25%).



(a) Smartphone

(b) Smartwatch

(c) Tablet

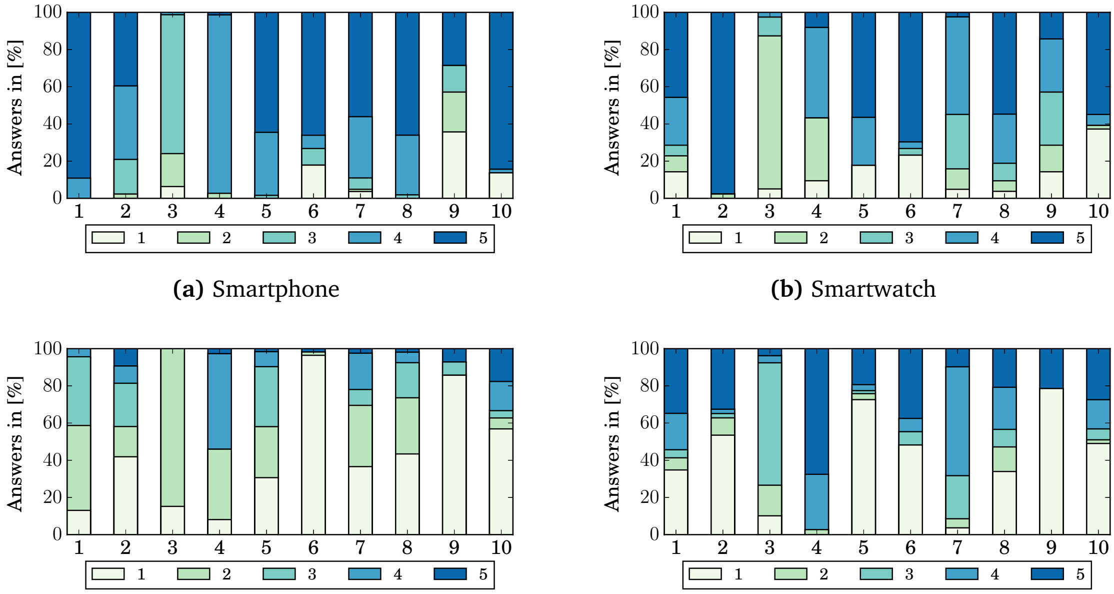
(d) PC

Figure 6.5.: Overview of answers to the questions about the proximity of the participant's devices. For each study participant it is shown how often he or she choose which value between 1 (far away) and 5 (easy reachable).

6.5. Device Proximity

The third question in the survey was about the proximity of the participant's devices to him- or herself. Each device was ranked with a 5-point Likert scale between the value 1 (far away) and 5 (easy reachable). Figure 6.5 shows the results for each study participant. It can be seen that smartphone and smartwatch were often in proximity to the participants. In 85.18% of the cases the participants ranked the smartphone with 4 or 5, followed by the smartwatch with 74.11% and the PC with 53.75%. The tablet only in 36.07% of the cases was ranked with 4 or 5. Although the PC is the device most often rated with 5 (43.04%) it also is rated in 34.29% of the cases with 1 or 2. Only the tablet is more often rated with 1 or 2 (40.71%). This may be a result of the fact that it in the case of a desktop PC is at a fixed position and that it is not as mobile as the smartphone and the smartwatch. If the participant sits in front of it and currently uses it, it is rated very near and if not, it is rated more far away because the participant has to walk to it and eventually first has to turn it on. On average the smartphone was rated to be in proximity the most often and the tablet the most often to be far away.

Moreover it can be seen that the proximity to devices strongly varies between participants. For example participant 9 is rarely close to the PC while participant 4 ranked the PC as "very near" in most answers.



(a) Smartphone **(b)** Smartwatch **(c)** Tablet **(d)** PC

Figure 6.6.: An overview of the answers to the question about where the study participants want to receive notifications on. For each study participant it is shown how often he or she choose which value between 1 (do not notify me on that device) and 5 (notify me on that device).

6.6. Which Device to Notify On

The last question of the survey was about on which device or devices the participants want to receive notifications on. Similar to the third question the study participants therefore had to rank each device with a 5-point Likert scale. The results for each of the participants are shown in Figure 6.6. Similar to question three (see Figure 6.5) the smartphone is most often the device the participant wants to receive a notification on. It is rated in 76.43% of the surveys with 4 or 5, followed by the smartwatch with 57.86% and the PC with 46.61%. On the tablet only 17.32% of the surveys are answered with 4 or 5. A reason for the high rating of the smartphone may be that participants mainly do messaging with the smartphone. Therefore the smartphone has established as a device for notifications. The results show that the tablet is the device with the lowest rating. In 70.71% of the surveys the tablet is ranked with a 1 or 2.

The results of this question also show, that the participants' preferences differ. For example participant 4 wants to have his notifications on the PC very often while participant 5 rarely wants notifications on the PC.

	in transit		at home		work/uni	
Device	M	SD	M	SD	M	SD
Phone	0.12	0.13	-0.18	0.21	0.13	0.31
Watch	0.04	0.25	-0.21	0.38	0.24	0.26
Tablet	-0.24	0.24	0.31	0.37	-0.22	0.26
PC	-0.35	0.10	0.21	0.31	0.24	0.33

Table 6.1.: Average correlation coefficient between the location of the study participants and on which device the notification should be shown on. The values are between -1 (negative correlation) and 1 (positive correlation).

6.7. Correlations between Questions of the Survey

One of the reasons for the study was to find the best device or devices for notifying the user. Therefore we have a look on correlations between questions of the survey and the last question about on which device the participants want to receive their notifications on.

First we analyzed the correlations between the question about the location of the participant and to which device the notification should be sent to. The average correlation coefficients for each location and device can be seen in Table 6.1. All calculated correlation coefficients for single participants can be seen in Table A.1-A.3. The coefficients are in a range from -1 to 1 and a greater absolute value represents a stronger positive or negative correlation. The correlation coefficients were calculated between binary values, for example the participant is at home or the participant is not at home, and the five-step signal resulting from the 5-point Likert scale, which the participant used to rank how likely he or she wants to receive a notification on the particular device. For three participants there are no correlation coefficients with being at *work/uni* because they did not answer this option in the survey. The results show that there are negative correlations between the participant is on transit and a notification should go to the tablet or to the PC. As a consequence this means that on average, participants does not want to get notified on the tablet or PC when they are *in transit*. Moreover the tablet has a negative correlation coefficient with *work/uni*. Smartphone and smartwatch have a negative correlation coefficient with *at home*. A possible reason for that can be the fact that smartphone and smartwatch at home more often are laying on a table or are charging. Thus the user may prefer one of the other devices. The smartwatch has a small correlation coefficient with *in transit* which means that being *in transit* has very low influence on whether the user wants a notification on the smartwatch. The tablet and the PC have positive correlation coefficients when being *at home*. This means that the participants more likely wanted to be notified on this devices when being *at home*. Moreover there are positive correlations for the smartphone when being *in transit* and for smartphone, smartwatch and PC when being *at work*.

Device	Participant										Total	
	1	2	3	4	5	6	7	8	9	10	M	SD
Phone	0.28	0.51	0.43	0.77	0.32	0.64	0.29	0.46	0.00	0.86	0.46	0.24
Watch	0.49	0.33	0.16	0.94	0.94	0.84	0.53	0.73	0.56	0.93	0.65	0.26
Tablet	0.58	0.78	0.52	0.90	0.86	0.85	0.28	0.68	0.52	0.62	0.66	0.18
PC	0.77	0.87	0.57	0.71	0.88	0.91	0.56	0.84	0.99	0.81	0.79	0.13

Table 6.2.: Correlation coefficient for each study participant between the proximity of the device and whether the notification should be shown on that device. The values are between -1 (negative correlation) and 1 (positive correlation).

In Table 6.2 the correlation coefficients between the questions about the proximity of a device to the participant and how likely the participant want to receive a notification on that device are shown for each study participant and each device. Both questions were answered with a 5-point Likert scale. It can be seen that there are no negative values in the table which means that if participants are near to a device, they more likely want to receive a notification on that device. However, the standard deviation is very high which indicates that preferences differ between single participants. The highest mean of the correlation coefficient has the PC (0.79) which may also be due the fact that it is not as mobile as the other devices. The mean correlation coefficient of the smartwatch is 0.65, followed by the tablet with 0.66. The average correlation coefficient of the smartphone is the lowest (0.46). A possible reason for that is that the smartphone is the device which is most often very near to the participants (as shown in Figure 6.5 (a)) and it is therefore not as meaningful as the answers to that question compared to the other devices.

6.8. Contextual Information

We also want to determine whether contextual information and sensor data obtained from the different devices can be used to find out which device or devices should be notified. Thus contextual data was recorded during the study. In this section we look at the *display on* event, the *interaction/touch* event and the *still* event from the *Google Play Services Location API*. These events were the most often triggered events we recorded and therefore significant results can be achieved. On the Android devices all of the mentioned events were logged, on the PC only the *interaction* event was recorded. To analyze the data, we computed for each answer of the survey whether the event is triggered at the same time or not. Because the smart devices were not synchronized and run independent from each other, the device time used for logging might be affected by clock drift. As a result we decided to find out whether the event was triggered in the time span from five minutes before the survey was completed to five minutes

	Participant										Total	
Device	1	2	3	4	5	6	7	8	9	10	M	SD
Phone	0.05	0.22	-0.02	0.03	0.09	0.38	0.36	-0.19	0.22	0.25	0.14	0.17
Watch	0.74	-0.08	0.14	0.25	0.50	0.23	0.28	0.19	0.58	0.65	0.35	0.25
Tablet		0.48	0.10	0.15	0.17	0.59	0.58	0.62		-0.11	0.32	0.26

Table 6.3.: Correlation coefficient for each study participant between the *display on* event and on which device the notification should be shown on. The values are between -1 (negative correlation) and 1 (positive correlation).

after the survey was completed. This also solves the issue that participants may interrupt what they were doing to answer the survey. For example it is possible that a participant is writing a message with the smartphone with display activated and when the survey triggers he or she may turn the display of the smartphone off to answer the survey. As a consequence the *display on* event would not be noticed when only considering a single point of time.

Table 6.3 shows the correlation coefficients for each participant and device between the *display on* event of a device and the value the last question, about whether the notification should go to this device or not, was rated with. There are no coefficients for the tablet for participant 1 and participant 9 because the display of the tablet never was enabled when answering the questions.

It can be seen that for most of the participants there is a positive correlation between the values. However, it varies between the participants. The largest average correlation coefficient has the smartwatch, followed by the tablet. The coefficient for the smartphone is the lowest. The average for each device is positive which means that when the *display on* event of a device was triggered, the participants more likely want to have notifications on that device. However, for some participants a negative correlation coefficient was measured which means that they do not want to receive a notification on the device the *display on* event was triggered.

On Table 6.4 the correlation coefficient between the *interaction* event of a device and the value of the last question for that device can be seen. On the PC the *interaction* event is triggered when a mouse movement or keyboard input occurs, on the Android devices the *interaction* event is triggered when the participant touches the activated screen. Because participant 4 did not touch the screen of the smartphone at the time of a survey and participant 1 and 9 did not touch the screen of the tablet at the time of a survey, there are no correlation coefficients in the table for the correspondent entries. On average the correlation coefficient for the PC is the highest and it is positive for each study participant. This means that the *interaction* event on the PC can be used to determine whether the notification should be sent to the PC or not. The correlation coefficient for the smartphone is the lowest and only slightly higher than the coefficient with the *display on* event. Moreover it can be seen that participant 3 and

	Participant										Total	
Device	1	2	3	4	5	6	7	8	9	10	M	SD
Phone	0.13	0.26	-0.01		0.04	0.35	0.35	-0.24	0.40	0.18	0.16	0.20
Watch	0.23	0.11	0.13	0.27	0.40	0.31	0.09	0.04	0.36	0.24	0.22	0.11
Tablet		0.57	0.07	0.21	0.23	0.97	0.62	0.51		-0.11	0.38	0.33
PC	0.45	0.90	0.43	0.41	0.78	0.24	0.32	0.42	0.60	0.34	0.49	0.20

Table 6.4.: Correlation coefficient for each study participant between the *Interaction/Touch* event and on which device the notification should be shown on. The values are between -1 (negative correlation) and 1 (positive correlation).

	Participant										Total	
Device	1	2	3	4	5	6	7	8	9	10	M	SD
Phone	-0.17	0.52	0.10	-0.01	-0.11		0.19	0.06		0.63	0.15	0.27
Watch	0.43		-0.24	0.21	0.64	0.32	0.30	0.13		0.48	0.28	0.25
Tablet		0.44			0.13	0.28				-0.14	0.18	0.21

Table 6.5.: Correlation coefficient for each study participant between the device is still and on which device the notification should be shown on. The values are between -1 (negative correlation) and 1 (positive correlation).

participant 8 had a negative correlation value with both: the *display on* and the *interaction* event. This may be a result of the fact that there is a relation of the two events because if the display is not activated there can not be a touch event either. On average each device has a positive correlation coefficient, which means that on average the participants want to receive notifications on a device a *touch* events were triggered. However, there were negative values for single participants, too, which means that these participants do not want to receive notifications on a device a *touch* event was triggered.

Table 6.5 shows the correlation coefficients between the state whether it is still or not and the rating value of the last question of that device. A device is considered as still if the *Google Play Services Location API* gives 100% confidence for the activity "still". This is achieved when the device is not moving, for example when it is on a table. It can be seen that Table 6.5 is more sparse than the previous ones, which is a result of the fact that it often occurs that a device is not moving all the time when a survey was answered. The average correlation coefficients for each device are positive. The highest correlation coefficient has the smartwatch with a mean of 0.28, followed by the tablet with 0.18 and the smartphone with 0.15.

6.9. Summary and Discussion

In this chapter we described the results of the study. We analyzed the dataset of ten participants which in total completed 560 surveys over a time period of seven days. The monitored devices were used in a different way. The smartwatch showed the most *display on* events but the total interaction time was lower than on the other devices. Moreover *touch* events on the smartwatch were very rare. The smartphone was used frequently and a number of *touch* events occurred on it. The tablet was used the least but for longer sessions and the PC was used for the longest total time. The participants answered most of the surveys *at home*, followed by *work/uni* and *in transit*. On average the smartphone was the device that was most often in proximity to the participant when completing a survey, followed by the smartwatch, the PC and the tablet. Moreover in the survey the devices were ranked on average in the same order for how preferably a notification should be sent to a device. A significant relation between the proximity of the devices to the participant and on which device the participant prefers the notification can also be seen at the correlation coefficients, which were positive for all study participants. The average coefficient was highest for the PC and lowest for the smartphone. In the previous section of this chapter we discussed correlations between which device or devices should receive notifications and contextual information at the same time. For the contextual information we considered the *display on* event, the *interaction* event and the *still* event, because these are the events which occurred very frequently. We found out that on average, there is a positive correlation between the events and the devices. However, this does not hold for all participants. Moreover some events were not triggered on some devices for some participants while answering a survey and therefore no correlation coefficient could be calculated in those cases. Nevertheless all the mentioned events on average correlate with where the notification should be forwarded to and therefore can be used to develop an algorithm for selecting the device or devices to notify the user.

Furthermore the results show that the proximity of a device to the user can work well to predict the device for notifying the user. However, the proximity can not be determined by looking at the events we currently record. Related work may be used to improve the determination of the proximity [WSB13]. Moreover there is more information such as additional events or the time since an event happened the last time which can be consulted for implementing a decision making algorithm.

The highest power for predicting where a notification should be sent to differs depending on the device. For the smartwatch the *display on* event had the highest correlation coefficient. For smartphone and tablet the *interaction* event produced the best results. For the PC only the *interaction* event was recorded but it has the highest total correlation coefficient which means that it should work well to determine whether the notification should be sent to that device. The least important event is the *still* event because it has small correlation coefficients and the devices are most of the time still.

7. Implementation of the Server Application

In this chapter we discuss the prototypical implementation of the server application for selecting the device or devices to notify the user. First we look at the overview of the server application and the client-server system. After that an exemplary algorithm based on the findings of the previous chapter, for deciding which device a notification should be forwarded to, is described.

7.1. Overview

In Chapter 5 the implementation of the client applications for Android and Windows have been described. These applications allow sending several events and sensor data to the server. To transfer data from the clients to the server, transmission control protocol (TCP) sockets are used. The server is implemented as a multi-threaded Java TCP server.

Because the server needs to keep track of the different devices and also has to distinguish them, a unique name is assigned to each device. The server stores the contextual information the clients transfer. As soon as contextual information changes, for example if the display of a device is turned on, the event and the time of the change will be transmitted to the server. The server receives the data and updates the corresponding fields. Therefore the server holds the last information of all the devices. This information can be used for decision making. Thus the server decides on the basis of the received data to which device a notification should be sent to. The prototypical implementation described in this thesis can hence be used for selecting the device. The distribution of notifications is not covered but can be implemented by adapting other multi-device notification frameworks.

7.2. Selecting the Device(s)

In the previous chapter we analyzed the results of the study (see Chapter 6). We found that users like to receive notifications on different devices depending on their surroundings. For example there are correlations between the proximity of a device to the user and how likely the user wants to receive a notification on this device. Because the aim of the server is to select a device or subset of devices for notifying the user, we therefore want to find a possibility to

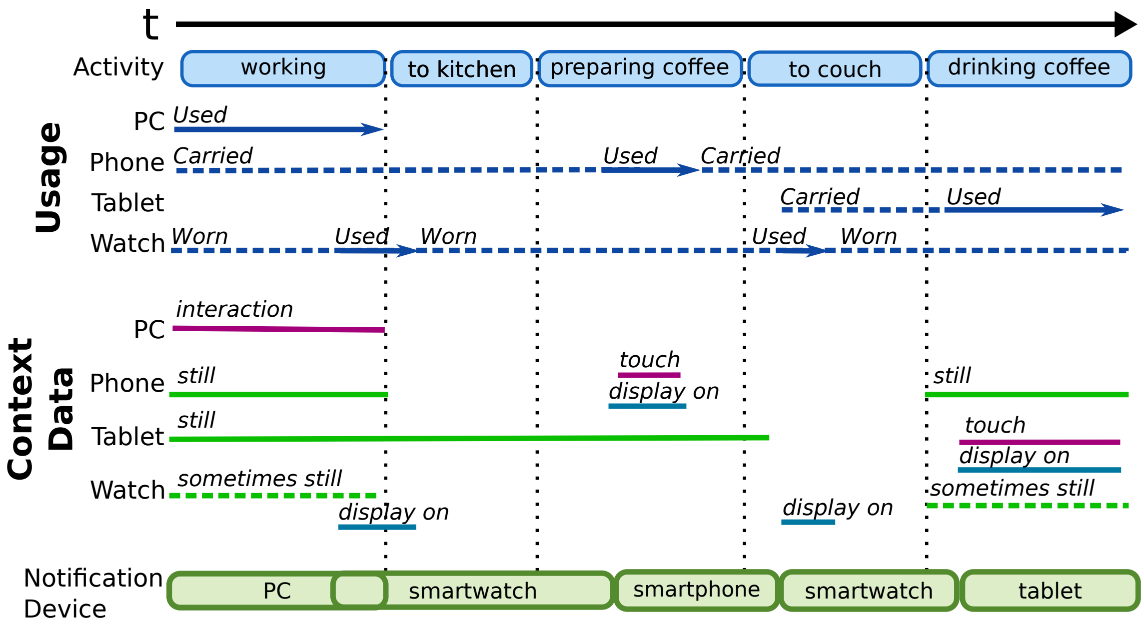


Figure 7.1.: An example scenario how devices could be used over time with context data and the result of a version of the device selection algorithm.

determine which device or devices are next to the user. Based on our findings the *display on* event, the *interaction* event and the *still* event can be used to determine a suitable device.

Thus the current approach is to check for each device d the time $t_{d,o}$ since the last *display on* event, the time $t_{d,i}$ since the last *interaction/touch* event and the time $t_{d,m}$ since the device was not still the last time. Now an evaluation function $f_d(t_{d,o}, t_{d,i}, t_{d,m}) = w_{d,o}t_{d,o} + w_{d,i}t_{d,i} + w_{d,m}t_{d,m}$ with the weights $w_{d,o}, w_{d,i}, w_{d,m} : w_{d,o} + w_{d,i} + w_{d,m} = 1$ can be used to determine a device ranking value. The PC is handled separately because the *interaction* event is the only event which is recorded. Thus the function for the PC is $f_{\text{PC}}(t_{\text{PC},i}) = t_{\text{PC},i}$. On the other devices the weights can be chosen according to the influences the corresponding events have on determining the device. Furthermore, it may be possible to compute suitable weights using a machine learning algorithm. If a notification is incoming, the server calculates the device ranking value for each device. Afterwards the device or devices with the smallest value can be selected for notifying the user. If the last event on all devices is more than an hour in the past, the information is not considered as up-to-date anymore and the server proposes to notify the user on all devices.

On Figure 7.1 the user's device usage scenario introduced in Chapter 3 is shown. Moreover contextual information like the *interaction*, the *display on*, the *still* event and which device the algorithm selects for notifying the user in the current contextual situation, can be seen. The weights for the devices are chosen such that only the event with the highest influence for predicting on which device a notification should be displayed are used. Thus for the smartwatch the *display on* event and for smartphone, tablet and PC the *interaction* events are considered. First the user is working on the PC and therefore the *interaction* event of the PC is triggered. As a consequence the user will be notified on that device. If the user uses the smartwatch, the *display on* event of the smartwatch is triggered and hence the ranking value of the PC is the same as the value of the smartwatch. Thus both devices can be selected for a notification at that time.

The significance of the selection may be improved by taking more events into account and by defining suitable weights for all events. Furthermore it would be possible to train machine learning algorithms from the data recorded during the study.

7.3. Summary

In this chapter we presented the implementation of the server application which is responsible to select the device or devices for notifying the user. The client applications running on the smart devices of the user collect contextual information and send it to the server. The server handles the data for each device. The device is selected by analyzing the last occurrences of the *display on* event, the *still* event and the *interaction* event. For each device a weight for the time since the last occurrence of an event is chosen. As seen in the previous chapter, the mentioned events are correlating for each device with the probability the user likes to receive a notification on the corresponding device.

8. Conclusion and Future Work

In this chapter the work of this thesis is summarized and conclusions based on our findings are drawn. At the end we present an outlook for future work.

8.1. Summary and Conclusions

In this thesis we introduced a concept for selecting a device or subset of devices for notifying the user. First we summarized previous work regarding notifications on different smart devices and their implementation. Notifications are an elementary part of interaction on smart devices and typically are viewed within minutes [PCO14]. Moreover studies have shown that notifications have a disruptive effect and can delay the execution of a task [CCH01, IH07, LBGK12]. In addition the disruptive effects are even exacerbated due to the fact that users have a growing number of smart devices in use. Nevertheless notifications are valued by users [IH10, PR15]. After that we presented how contextual information from smart devices can already be used for determining the position of smartphones [WSB13] or for delaying the delivery of notifications [FGB11, HI05, IB08, ORN+15, SD14]. Afterwards we had a look at current implementations of multi-device notification frameworks. Based on the related work described, we came to the conclusion that there is the necessity to identify the right balance between keeping the user informed and limit the negative effects. Thus we wanted to find the most suitable device or subset of devices to notify the user, based on contextual information obtained from the user's smart devices.

To this aim we developed a framework with the capability of forwarding contextual information from the client applications to the server. As a consequence the server is able to run a centralized algorithm for deciding to which device a notification should be forwarded. For the implementation of the framework we started to develop applications for the client side. We developed an Android application running on smartphones, tablets and smartwatches and a Windows Application for desktop PCs and notebooks. These applications are collecting several types of contextual information and system events such as display state, interaction events, Wi-Fi state and acceleration data. This data can be cached locally on the client or sent to the server.

To analyze which part of the collected data is suitable for selecting a device or subset of devices to notify the user, we conducted a study before the actual implementation of the server application. The aim of the study was to obtain a better understanding of how notifications could be distributed across multiple devices in a smart way. Therefore we used the Experience Sampling Method to periodically interrupt the participant's daily life and opened a survey about the participant's surroundings, the proximity of his or her devices and on which device he or she desires to receive a notification. Furthermore the applications to collect contextual information were installed on the devices of the study participants. This allowed us to draw conclusions about the relation between the contextual information from the devices and on which device the participants like to receive the notification.

In total we had 10 study participants completing 560 surveys over a seven day period. Using the data from the study we first analyzed how the study participants used their devices. The smartphone and the smartwatch were used for shorter sessions but were used more frequently while PC and tablet were used for longer sessions. The PC was the device the most often used. Nevertheless, the smartphone was the device that was most often nearby the user when answering a survey. Also the smartphone was the device the users most often desired to receive their notifications on. Thus in 76% of the cases it was rated with 4 or 5 on a 5-point Likert scale.

Furthermore we found out that the proximity of a device strongly correlates with whether a notification should be sent to that device. This can be seen at the positive correlation coefficient for all participants and all devices. Moreover we analyzed for each device whether the *display on* event, the *interaction* event and the *still* event are correlating with how likely a notification should be sent to that device. We found out that on average there is a positive correlation for each device but not for every participant. This means that users' preferences differ and therefore no holistic algorithm working adequately well for every user can be found. Based on our findings we implemented the server to select a device depending on the weighted data obtained from the client devices. As a result the negative effects of notifications in a multi-device environment could be minimized.

8.2. Future Work

The decision making algorithm of the server is based on the previous study. For further investigations a follow-up study for evaluating the implementation would be favorable. Beyond that, the algorithm can also be improved by considering more sensor events and more contextual data. Actually only a fraction of the theoretically available data is used in our studies, which is a good starting point for further investigations. To improve the statistics of the results for the events which occur rarely, much more participants or a study over a longer period of time per participant would be desireable, but is beyond the scope of this thesis.

Moreover there is the possibility to obtain more contextual information from additional sources to improve the decision making algorithm. For example the user's calendar can be parsed. Another way to improve the decision making algorithm can be the use of machine learning methods. However, there is the necessity for more training data to get sufficient results, too. Because users' preferences differ, there is also the possibility to let users train their selection algorithm individually. For example this can be achieved by first spreading notifications to all devices and then analyzing the time the user needs to react to the notification on a certain device in order to train the algorithm.

Because the amount of devices which are able to display notifications increases, it is possible to adapt the framework for a variable amount of devices. Such devices can vary in sensors and events from the already supported smart devices. For example the framework could be adapted for smart TVs or smart lamps in the future.

In the current implementation mainly contextual information obtained from the devices is used for decision making. However, the content of a notification can also be considered to improve the decision for notifying the user the user. For example an important notification could be distributed across all devices while other notifications could be sent to a single device to minimize the effect of disruption.

Related work showed that there is the possibility to use the contextual information to delay a notification until an appropriate moment occurs [FGB11, HI05, IB08, ORN$^+$15, SD14] or to change the intensity of the alert [SD14]. These techniques may be combined with the selection of a device or subset of devices for a notification. For example a notification could be delayed until a device reaches an acceptable point for a notification. Then the notification could be sent to that device. Based on that an already existing multi-device notification framework, for example the one by [WSH15] and the one proposed in this thesis, may be extended to build a holistic notification management system, routing notifications to devices based on multiple input variables and contextual information. Concludingly this work can be seen as profound basis for further investigations towards an intelligent notification system in smart multi-device computing environments.

A. Appendix

Correlation coefficients between the first and the last question of the survey for each participant.

	Participant										Total	
Device	1	2	3	4	5	6	7	8	9	10	M	SD
Phone	0.16	0.18	-0.06	0.01	0.14	-0.01	0.08	0.06	0.43	0.18	0.12	0.13
Watch	0.19	0.08	-0.03	0.08	-0.64	0.10	0.20	0.01	0.31	0.14	0.04	0.25
Tablet	-0.35	-0.49	-0.38	0.20	-0.67	-0.06	-0.11	-0.19	-0.24	-0.05	-0.24	0.24
PC	-0.36	-0.43	-0.34	-0.49	-0.34	-0.38	-0.09	-0.33	-0.33	-0.38	-0.35	0.10

Table A.1.: Correlation coefficient for each study participant between being *in transit* and on which device the notification should be shown on. The values are between -1 (negative correlation) and 1 (positive correlation).

	Participant										Total	
Device	1	2	3	4	5	6	7	8	9	10	M	SD
Phone	-0.27	-0.43	0.17	-0.01	-0.09	-0.33	-0.17	0.13	-0.28	-0.49	-0.18	0.21
Watch	-0.79	-0.10	0.01	-0.08	0.60	-0.47	-0.18	-0.30	-0.03	-0.74	-0.21	0.38
Tablet	0.33	0.63	0.85	-0.20	0.70	0.34	0.25	0.54	-0.11	-0.25	0.31	0.37
PC	0.06	0.55	0.27	0.49	0.36	-0.38	0.61	0.36	-0.14	-0.07	0.21	0.31

Table A.2.: Correlation coefficient for each study participant between being *at home* and on which device the notification should be shown on. The values are between -1 (negative correlation) and 1 (positive correlation).

	Participant										Total	
Device	1	2	3	4	5	6	7	8	9	10	M	SD
Phone	0.17	0.34	0.22			0.47		0.05	-0.57	0.23	0.13	0.31
Watch	0.53	0.06	-0.06			0.48		0.23	-0.09	0.54	0.24	0.26
Tablet	0.00	-0.39	-0.54			-0.28		-0.36	-0.29	0.28	-0.22	0.26
PC	0.31	-0.33	0.14			0.57		-0.06	0.70	0.35	0.24	0.33

Table A.3.: Correlation coefficient for each study participant between being *work/uni* and on which device the notification should be shown on. The values are between -1 (negative correlation) and 1 (positive correlation). Empty entries are due the fact that the regarding participant weren't on work/uni during the time of the answered surveys.

Bibliography

[ABBE08] R. M. Arlein, S. Betge-Brezetz, J. R. Ensor. Adaptive notification framework for converged environments. *Bell Labs Technical Journal*, 13(2):155–159, 2008. (Cited on page 18)

[BK06] B. P. Bailey, J. A. Konstan. On the need for attention-aware systems: Measuring effects of interruption on task performance, error rate, and affective state. *Computers in human behavior*, 22(4):685–708, 2006. (Cited on page 17)

[BLG+14] M. Böhmer, C. Lander, S. Gehring, D. P. Brumby, A. Krüger. Interrupted by a phone call: exploring designs for lowering the impact of call notifications for smartphone users. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 3045–3054. ACM, 2014. (Cited on page 16)

[CCH01] E. Cutrell, M. Czerwinski, E. Horvitz. Notification, disruption, and memory: Effects of messaging interruptions on memory and performance. 2001. (Cited on pages 16 and 51)

[CL87] M. Csikszentmihalyi, R. Larson. Validity and reliability of the Experience-Sampling Method. *The Journal of nervous and mental disease*, 175(9):526–536, 1987. (Cited on page 11)

[DWF+11] A. K. Dey, K. Wac, D. Ferreira, K. Tassini, J.-H. Hong, J. Ramos. Getting closer: an empirical investigation of the proximity of user to their smart phones. In *Proceedings of the 13th international conference on Ubiquitous computing*, pp. 163–172. ACM, 2011. (Cited on page 16)

[FEW12] A. P. Felt, S. Egelman, D. Wagner. I've got 99 problems, but vibration ain't one: a survey of smartphone users' concerns. In *Proceedings of the second ACM workshop on Security and privacy in smartphones and mobile devices*, pp. 33–44. ACM, 2012. (Cited on page 27)

[FGB11] J. E. Fischer, C. Greenhalgh, S. Benford. Investigating episodes of mobile phone activity as indicators of opportune moments to deliver notifications. In *Proceedings of the 13th international conference on human computer interaction with mobile devices and services*, pp. 181–190. ACM, 2011. (Cited on pages 16, 51 and 53)

[FGK+14] D. Ferreira, J. Goncalves, V. Kostakos, L. Barkhuus, A. K. Dey. Contextual experience sampling of mobile application micro-usage. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*, pp. 91–100. ACM, 2014. (Cited on pages 15 and 38)

[FMK+10] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, D. Estrin. Diversity in smartphone usage. In *Proceedings of the 8th international conference on Mobile systems, applications, and services*, pp. 179–194. ACM, 2010. (Cited on page 38)

[goo] Google Inc. Making Your App Location-Aware. URL https://developer.android.com/training/location/index.html. (Cited on page 28)

[HI05] J. Ho, S. S. Intille. Using context-aware computing to reduce the perceived burden of interruptions from mobile devices. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 909–918. ACM, 2005. (Cited on pages 16, 17, 19, 51 and 53)

[HKPH03] E. Horvitz, C. Kadie, T. Paek, D. Hovel. Models of attention in computing and communication: from principles to applications. *Communications of the ACM*, 46(3):52–59, 2003. (Cited on page 18)

[HL00] R. Hansson, P. Ljungstrand. The reminder bracelet: subtle notification cues for mobile devices. In *CHI'00 Extended Abstracts on Human Factors in Computing Systems*, pp. 323–324. ACM, 2000. (Cited on page 16)

[IB08] S. T. Iqbal, B. P. Bailey. Effects of intelligent notification management on users and their tasks. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 93–102. ACM, 2008. (Cited on pages 16, 17, 19, 51 and 53)

[IH07] S. T. Iqbal, E. Horvitz. Disruption and recovery of computing tasks: field study, analysis, and directions. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pp. 677–686. ACM, 2007. (Cited on pages 11, 16, 18 and 51)

[IH10] S. T. Iqbal, E. Horvitz. Notifications and awareness: a field study of alert usage and preferences. In *Proceedings of the 2010 ACM conference on Computer supported cooperative work*, pp. 27–30. ACM, 2010. (Cited on pages 16 and 51)

[KSSHS13] J. Knittel, A. Sahami Shirazi, N. Henze, A. Schmidt. Utilizing contextual information for mobile communication. In *CHI'13 Extended Abstracts on Human Factors in Computing Systems*, pp. 1371–1376. ACM, 2013. (Cited on page 16)

[LBGK12] L. Leiva, M. Böhmer, S. Gehring, A. Krüger. Back to the app: the costs of mobile application interruptions. In *Proceedings of the 14th international conference on Human-computer interaction with mobile devices and services*, pp. 291–294. ACM, 2012. (Cited on pages 11, 16, 18 and 51)

[MPO13] H. Müller, M. Pielot, R. de Oliveira. Towards Ambient Notifications. *Peripheral Interaction: Embedding HCI in Everyday Life*, p. 21, 2013. (Cited on page 15)

[NMS15] L. Norrie, R. Murray-Smith. Impact of Smartphone Notification Display Choice in a Typing Task. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, pp. 1094–1099. ACM, 2015. (Cited on page 16)

[ORN+15] T. Okoshi, J. Ramos, H. Nozaki, J. Nakazawa, A. K. Dey, H. Tokuda. Reducing users' perceived mental effort due to interruptive notifications in multi-device mobile environments. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pp. 475–486. ACM, 2015. (Cited on pages 16, 17, 51 and 53)

[PCO14] M. Pielot, K. Church, R. de Oliveira. An in-situ study of mobile phone notifications. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services*, pp. 233–242. ACM, 2014. (Cited on pages 15 and 51)

[PR15] M. Pielot, L. Rello. The Do Not Disturb Challenge: A Day Without Notifications. In *Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems*, pp. 1761–1766. ACM, 2015. (Cited on pages 11, 16, 18, 22 and 51)

[SD14] J. Smith, N. Dulay. Ringlearn: Long-term mitigation of disruptive smartphone interruptions. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on*, pp. 27–35. IEEE, 2014. (Cited on pages 16, 17, 51 and 53)

[SH15] A. S. Shirazi, N. Henze. Assessment of Notifications on Smartwatches. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, pp. 1111–1116. ACM, 2015. (Cited on page 15)

[SSHD+14] A. Sahami Shirazi, N. Henze, T. Dingler, M. Pielot, D. Weber, A. Schmidt. Large-scale assessment of mobile notifications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 3055–3064. ACM, 2014. (Cited on pages 15, 16 and 18)

[WSB13] J. Wiese, T. S. Saponas, A. Brush. Phoneprioception: enabling mobile phones to infer where they are kept. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2157–2166. ACM, 2013. (Cited on pages 17, 19, 46 and 51)

[WSH15] D. Weber, A. S. Shirazi, N. Henze. Towards Smart Notifications using Research in the Large. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, pp. 1117–1122. ACM, 2015. (Cited on pages 18, 24 and 53)

All links were last followed on October 30, 2015.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

place, date, signature