Institute for Visualization and Interactive Systems

University of Stuttgart
Universitätsstraße 38
D–70569 Stuttgart

Master Thesis Nr. 106

Notification Strategies in Smart Environments

Florian Gänßlen

Course of Study:	Softwaretechnik
Examiner:	Jun.-Prof. Dr. Niels Henze
Supervisor:	Dipl. Inf. Alexandra Voit, Thomas Kubitza M.Sc.
Commenced:	May 24, 2016
Completed:	November 23, 2016
CR-Classification:	H.5.2

Abstract

In the field of mobile devices, notifications represent a simple way to inform the users about new emails, messages or incoming phone calls. Besides the display, different output channels can be used to reinforce the hint that a new information has arrived. Simple melodies can be played by speakers or small electric motors can vibrate in different patterns to transport an information. Over time, the users learn how to distinguish these channels and which melody or vibration pattern represents a new email or an incoming phone call. But since 2007, when the company Apple introduced the first smart phone (known as the iPhone), the whole system, how notifications are presented to the user, didn't change much. The main restriction is, that new notifications are bound to the device to which they arrive. Every time a new notification is posted by an app, the device rings or vibrates and the user has to take it out of his pocket or look for it in his near environment. In some cases, this can be very impractical and annoying when it happens in the middle of a conversation or the user is interrupted while he was working on something. After the user inspected the notification about a new arrived email or message, the new information turns out to be unimportant. If the user had the possibility to be informed about the new notification by another display or device that is near him, he could take a quick look and see if it is something important or not and then return to his previous activity. This thesis presents a notification system that displays user notifications from Android smart phones on different devices that are connected to a central platform. The output devices are simple tablet computers and controllable bulbs, so that they can easily and ambient be integrated in a home environment. We developed different methods to present users their notifications and evaluated them together with the notification system in a user study. In general, we wanted to find out in which situations do users prefer which output devices or methods for which notification types. The gained insights can be used to improve existing notification systems or develop new strategies how notifications could be presented to users.

Kurzfassung

Im Bereich der mobilen Geräte stellen Benachrichtigungen eine einfache Möglichkeit dar, um die Benutzer über neue E-Mails, Text-Nachrichten oder eingehende Anrufe zu informieren. Neben dem Gerätedisplay können dabei noch weitere Ausgabekanäle verwendet werden, um den Benutzer auf eine neue Nachricht aufmerksam zu machen. Einfache Melodien können über Lautsprecher wiedergegeben werden oder kleine elektrische Motoren können verschiedene Vibrationsmuster verwenden, um eine Information zu transportieren. Im Laufe der Zeit lernen die Benutzer diese verschiedenen Kanäle voneinander zu unterscheiden und welche Melodie oder welches Vibrationsmuster eine neue E-Mail oder einen eingehenden Anruf darstellt. Aber seit dem Jahr 2007, als das Unternehmen Apple das erste Smartphone (bekannt als das iPhone) vorgestellt hat, gab es keine größeren Änderungen an der Art und Weise wie Benachrichtigungen den Nutzern präsentiert werden. Die größte Beschränkung liegt nach wie vor darin, dass eingehende Benachrichtigungen an das Gerät gebunden sind auf welchem sie ankommen. Jedes Mal, wenn eine neue Benachrichtigung von einer installierten App erzeugt wurde, klingelt oder vibriert das Gerät und der Benutzer muss es entweder aus seiner Tasche nehmen oder in seiner Umgebung suchen. In einigen Situationen kann dies sehr unpraktisch und störend sein, beispielsweise wenn dies während einer Unterhaltung geschieht oder der Benutzer unterbrochen wurde, während er an etwas gearbeitet hat. Nachdem der Benutzer dann die neu eingegangene Nachricht näher betrachtet hat, so erweist sich die neue Information als unwichtig. Hätte der Benutzer die Möglichkeit gehabt, die Nachricht auf einem nahestehenden Gerät oder Display zu betrachten, so hätte er einen flüchtigen Blick darauf werfen können, um zu beurteilen, ob es etwas Wichtiges ist oder nicht und seine vorherige Aktivität wiederaufnehmen. Diese Arbeit stellt ein Benachrichtigungssystem vor, das eingehende Benachrichtigungen von einem Android-Smartphone auf unterschiedlichen Geräten anzeigen kann, welche mit einer zentralen Plattform verbunden sind. Die Ausgabegeräte können einfache Tabletcomputer oder ansteuerbare Glühbirnen sein, so dass sie sich einfach und unaufdringlich in eine Heimumgebung integrieren lassen. Für die Darstellung der Benutzer-Benachrichtigungen wurden verschiedene Ausgabemethoden entwickelt und zusammen mit dem Benachrichtigungssystem in einer Benutzer-Studie ausgewertet. Grundsätzlich sollte herausgefunden werden, in welchen Situationen die Benutzer welche Ausgabegeräte- oder Methoden für welche Nachrichtentypen bevorzugen. Die gewonnenen Erkenntnisse können dazu verwendet werden, bestehende Benachrichtigungssysteme zu verbessern oder neue Strategien zu entwickeln, wie Benachrichtigungen den Benutzern in Zukunft präsentiert werden könnten.

List of Abbreviations

BLE Bluetooth Low Energy

GPS Global Positioning System

HTTP Hyper Text Transfer Protocol

IOT Internet of Things

JSON JavaScript Object Notation

LED Light-Emitting Diode

RSS Received Signal Strength

SIP Session Initiation Protocol

SMS Short Message Service

UDP User Datagram Protocol

URL Uniform Resource Locator

Contents

List of Figures

List of Tables

List of Listings

1 Introduction

In the field of mobile devices, notifications represent a simple way to inform the users about new emails, messages or incoming phone calls. Besides the display, different output channels can be used to reinforce the hint that a new information has arrived. Simple melodies can be played by speakers or small electric motors can vibrate in different patterns to transport an information. Over time, the users learn how to distinguish these channels and which melody or vibration pattern represents a new email or an incoming phone call.

But since 2007, when the company Apple introduced the first smart phone (known as the iPhone), the whole system, how notifications are presented to the user, didn't change much. The main restriction is, that new notifications are bound to the device to which they arrive. Every time a new notification is posted by an app, the device rings or vibrates and the user has to take it out of his pocket or look for it in his near environment. In some cases, this can be very impractical and annoying when it happens in the middle of a conversation or the user is interrupted while he was working on something. After the user inspected the notification about a new arrived email or message, the new information turns out to be unimportant. If the user would had the possibility to be informed about the new notification by another display or device that is near him, he could take a quick look and see if it is something important or not and then return to his previous activity.

Previous work tried to integrate different devices into the environment of the users and used them to display and visualize information and notifications. Popular examples of these *Ambient Information Systems* can be found from Weiser [40], Miller [23] or Dahley [9]. The so-called Internet of Things (IoT), which has gained increasing popularity over the last few years, allows simple household appliances such as washing or coffee machines to be integrated into an existing network, so that the users can be notified about their current state on a PC or smart phone. In 2014 Pielot et al. [26] conducted a user study in which they found out, that each of their participants had to deal with over 60 notifications on average per day. When we think about this number and all the different systems and devices that will come in the future, it is very likely that the number of notifications per day will increase. So, if we don't want to be overwhelmed by them at one day, we need new systems and ways to handle them.

This thesis presents a notification system that displays user notifications from Android smart phones on different devices that are connected to the *meSchup* platform [19]. The output devices are simple tablet computers and controllable bulbs, so that they can easily and ambient be integrated in a home environment. We developed different methods to present users their notifications and evaluated them together with the notification system in a user study. In general, we wanted to find out in which situations do users prefer which output devices or methods for which notification types. The gained insights can be used to improve existing notification systems or develop new strategies how notifications could be presented to users.

Structure

Chapter 2 gives an overview about the related work regarding notifications and their usage in different systems. The following chapter 3 will present a concept for a notification system that can be used in home scenarios. Details about the implementation can be found in chapter 4. The results of our user study that evaluated the system are presented in chapter 5. The final chapter 6 will summarize the results of this thesis and propose ideas for future work.

2 Related Work

In this chapter we will discuss the related work regarding notifications and ambient information systems. The first section presents previous work about notifications and their possible interruptive effect. We will then look at notifications and their presentation on different devices. The third section is all about ambient information systems with various software and hardware examples. The final and last section discusses smart environments and notification systems.

2.1 Notifications and Interruptions

The following user studies from various researchers focused on the interruptive effect of notifications while working on a task. Iqbal and Bailey [16] defined a notification as "the proactive delivery of information to a user." Notifications can be presented as "a visual cue, auditory signal, or haptic alert generated by an application." They also mentioned, that notifications can lead to interruptions and decrease the productivity.

Czerwinski et al. [8] performed a diary study over a week, in which the participants had to record and categorize their daily tasks, the number of interruptions and how difficult it was to resume a task after an interruption. Most of the interruptions were self-initiated or caused by telephone calls or appointments. The evaluation of the diaries showed, that returning to an interrupted task was rated as difficult and that longer tasks experienced more interruptions.

Bailey and Konstan [4] conducted a user study, in which the participants of an experimental group had to solve different tasks, like counting the occur of a specific word in a list of words. From time to time they were interrupted during this task with a second task, e.g. reading a news article and answering a question related to it. A control group did the same tasks but one after another. The results showed, that users who were interrupted during a task, completed them slower than the users from the control group that were not interrupted.

Iqbal and Horvitz [17] recorded in a field study the interactions of their participants with an email program and studied how disruptive notifications (or their absence) can be.

The results showed, that the majority of notifications did not cause the users to interrupt their current task and switch to the new email. After disabling the notification option, some users became even more focused and productive with their ongoing task, while other users spent more time accessing the email program on their own to look for new emails. In summary, the users viewed notifications “as a mechanism to provide passive awareness rather than a trigger to switch tasks.” The users were willing to accept the potential interruptions, because they thought, that the awareness effect of notifications is more valuable.

2.2 Mobile Notifications

To inform a person that something has happened, the devices that surround us every day use different channels to gain our attention. Visual (blinking lights), auditory (a ring tone is played) or tactile (vibrations) cues can be used simultaneously or individually to enforce the arrival of a new notification. In 2002, Hansson et al. [14] developed a model that tries to identify “the best” information cue in a certain situation. They thought, that the cues are “attention demanding, distinct and intrusive and therefore be perceived as inappropriate in many social situations”, for example when during a conversation of two people the phone of one person vibrates, while the other person has no idea what is going on.

Weber et al. [39] developed the Android-based *Desktop Notification* system that shares notifications in a multi-device environment. Users can display their smart phone notifications on tablets, PCs or Smart TVs. An app sends notifications to a server, which knows the user-device-relations. The server pushes notifications to a message service from Google, that forwards the notification to all connected devices. Via this system, Sahami Shirazi et al. [29] collected nearly 200 million notifications from more than 40,000 users over a time of 8 months and analyzed them. They published an application in the Google Play Store that forwards the notifications from the smart phone to a desktop PC and displays them in a web browser as small popup windows. Their results showed, that messenger, mail and calendar apps, like WhatsApp, Gmail or the Google Calendar, posted the most notifications per day and that these app categories were also rated as the most important ones. Based on this findings they concluded, that “important notifications are about people and events.” System and music notifications were rated as the most unimportant ones, because they are created too frequently and in some cases they “flood” the user with information.

Pielot et al. [26] conducted a user study in which they found out, that their participants had to deal with over 60 notifications per day. On every smart phone they installed a logging app that collected details about the posted notifications. Their data showed, that messaging apps like WhatsApp or GoogleTalk posted the most notifications, followed by email and social apps like Facebook.

2.2.1 Notifications on other Devices

Weber et al. [38] developed different design guidelines, how mobile notifications could be displayed on Smart TVs. Within three focus groups, they collected various design ideas, like the *toast notification* and the *ticker style,* which can also be found on other desktop and mobile systems. During their meetings, the participants expressed different concerns regarding the designs. No one wanted to be glared by a bright notification while watching a movie in a dark room. They also suggested a “family mode” that recognizes if there are other people in front of the TV. The notifications will then be disabled completely or show only the name of the sender and no parts of the message. Via an online survey, they asked potential users which design ideas they would prefer. The results showed, that only a generic icon doesn’t provide enough information about the notification for the participants. Therefore, they would like to see at least the name of the sender and a small excerpt of the notification text. Via a lab study they wanted to find out, how each user wants to display the notifications on the TV screen. They developed a smart TV application that was able to display notifications while watching TV. The users could change the position, the size, the opacity and the duration how long each notification should be displayed. The results showed, that users that are alone in front of the TV, have different preferences then users that were in the room with a researcher. All participants disabled the sound for new notifications, with the arguments that it is distracting or unnecessary. In summary, the paper concludes that developers should consider the priority of each notification and provide different customization options for users.

Voit et al. [36] asked in an online survey where and in which scenario users would like to receive IoT notifications in a smart home environment. Users had to rate the locations *smartphone*, *display*, *object* and *body* in the scenarios *entrance door, closing a window*, *taking medicine* and *watering a flower*. The results showed, that the participants preferred notifications on a smart phone more than on-body devices (“easily perceptible but also distracting”) while notifications on the display or on other objects were rated as difficult to perceive.

2.3 Ambient Information Systems

In this section, we want to explain the term *Ambient Information System* how it is described in the related work. After that, we present various hardware and software examples of ambient systems that were designed and developed over the last years. Light systems actually belong in the category of hardware systems, but since the related literature is very comprehensive, we list them in a separate section.

2.3.1 General

One of the first ideas to present and visualize information on another device than the screen of desktop computer, was described in 1995 by Weiser and Brown [40]. The so called *Dangling String*, created by the artist Natalie Jermijenko, was a small plastic string mounted at a ceiling and moved by a small electric motor. The motor was connected with an Ethernet cable and each bit of information caused a tiny twitch of the motor. A very busy network caused a jumping string with another noise than a quiet network that causes only a small twitch every seconds. They called this design "calm technology", because they saw it as a simple and smooth way to present information that can also be seen and heard by multiple people at the same time without disturbing their current activity.

In the following years, many researchers developed hardware and software systems to visualize information on devices that are integrated in the environment or periphery of the user. Many different terms haven been used to describe such systems, for example *Peripheral Displays*, *Ambient Displays* or *Notification Systems* [28]. Wiesneski et al. [41] described the term Ambient Display as the following: "information is moved off the screen into the physical environment, manifesting itself as subtle changes in form, movement, sound, color, smell, temperature, or light. Ambient displays are well suited as a means to keep users aware of people or general states of large systems, like network traffic and weather." For Matthews et al. [20] Peripheral Displays "allow a person to be aware of information while she is attending to some other primary task or activity." They also defined a toolkit that should help designers to create new peripheral displays by identifying and describing three important characteristics: *abstraction* and *notification* levels as well as different *transitions* from on notification to another. In their analysis of existing peripheral displays they found this subjects frequently and described them in detail.

Pousman and Stasko [28] proposed the term *ambient information system* for these kind of systems and characterized them, that they "display information that is important but not critical" and that they "can move from the periphery to the focus of attention

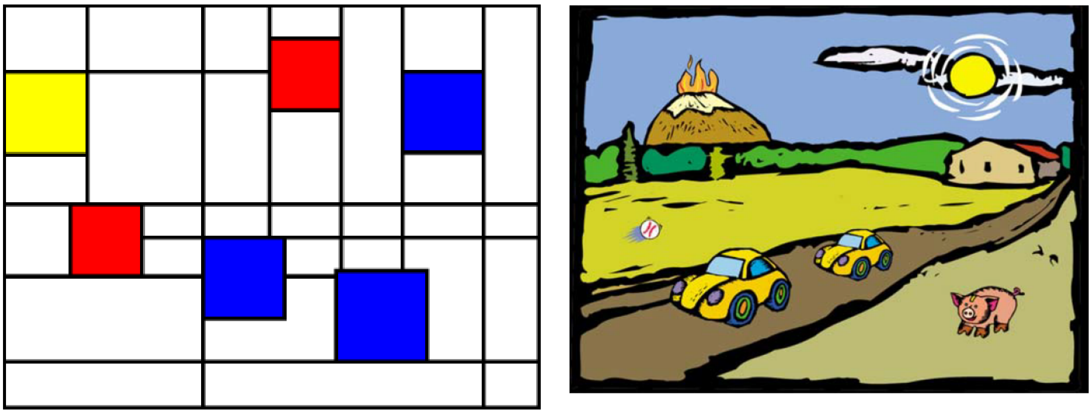


Figure 2.1: *Left:* Mondrian-like visualizations [31]. *Right:* InfoCanvas [23].

and back again." The authors analyzed 19 Ambient Systems, like the *Dangling String* [40], *InfoCanvas* [23] or the *Water Lamp* [9], and detected four properties that are shared by all systems. For each system, they rated these four properties from low to high, and identified one of four possible design patterns. The four properties were called *information capacity* (how much information can be displayed at the same time), *notification level* (to which degree the user is interrupted), *representational fidelity* (how accurate can the system display the information) and *aesthetic emphasis* (how does the system look, how good can it be integrated into an environment).

Compared to Pousman and Stasko, Tomitsch et al. [32] used the categories *abstraction level*, *transition*, *notification level*, *temporal gradient*, *representation*, *modality*, *source*, *location* and *dynamic of input* for different ambient information systems to "provide a balance between simplicity and descriptive power."

2.3.2 Software Systems

In the area of ambient information visualizations, Holmquist and Skog [31] tried to integrate informative art in an everyday environment. They developed a system that produces artworks based on the work of the Dutch artist Piet Mondrian, to display different types of information like the current e-mail traffic, weather forecasts or bus departure times. Mondrian's paintings are very characteristic and immediately recognizable because he used only three colors: red, yellow and blue. Holmquist and Skog used this color scheme and the geometrical shapes to map their data to the three parameters size, position and color. For a weather forecast a yellow color means sunny or clear, blue means rain or snow and red means overcast whereas the size of a square stands for the estimated temperature in Celsius.

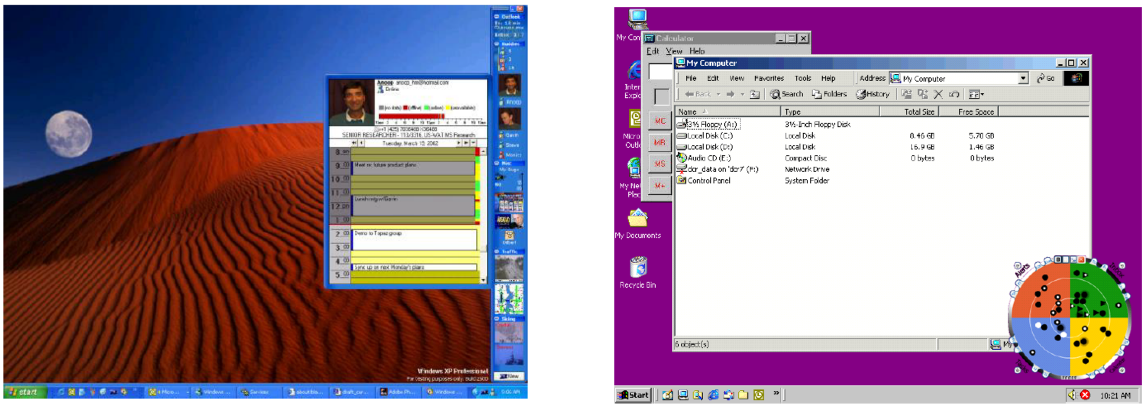


Figure 2.2: *Left:* Sideshow [5]. *Right:* Scope [34].

The major problem with this type of information visualization is, that it's not very intuitive. First, the viewer has to learn what each parameter represents in the visualization. There can always be a small description next to the painting, but the goal should be that everything is self-explaining. Second, it's difficult to translate the size of a square to an accurate measure in the real world. For the amount of remaining minutes until the next bus arrives this translation could work, but it will become much more difficult to map the size of a square to a concrete temperature.

Another visualization software system from Miller and Stasko [23] displays various information sources in form of a painting on a virtual canvas. *InfoCanvas* consists of different components that can can be used to display weather forecasts, news, stock quotes or email notifications. The user can fully customize the painting and map small images to the information sources. For the daily stock market performance the user can use the sky or the weather, "clear skies meaning good performance, or a storm indicating that the markets are down." The local traffic conditions can be mapped to the number of cars in the painting and when an important mail has been received, a volcano can erupt in the background.

Sideshow from Cadiz et al. [5] is a sidebar that tries to display important information, like upcoming meetings or appointments, new emails or notifications, the current status of a person, the weather forecast or the current traffic status. Because it's sticked to one side of the screen, it's always "visually persistent in people's periphery when they are working on their computer." *Scope* from van Dantzich et al. [34] combines the features from Sideshow with a more simplistic design, that can also display the importance of a task or calendar event: the closer a notification to the center of the circle is, the more important it is.

2.3.3 Hardware Systems

Dahley et al. [9] developed different devices, so called *Ambient Fixtures*, that can be placed anywhere in a room and are used as external displays for different types of information. The first device was called *water lamp* and used a computer-controlled solenoid that produced ripples by tapping the water in a pan. A lamp, which shines through that pan, projects the changing patterns of light and shadow onto a ceiling. The second device were simple *pinwheels* which should "explore the ideas of physical movement caused by invisible information flow. The pinwheels spin in the 'bit wind' at different speeds based upon their input information source." The authors imagined different application for their devices, for example the water lamp could display the heartbeat of another person, picked up by a special wristwatch, the current wireless LAN traffic or the seismographic activities in an area, that is vulnerable for earthquakes.

AuraOrb from Altosaar et al. [2] is a hardware orb, that uses LED lights and a text display to present notifications. With the arrival of the first notification, the orb starts showing an ambient light, which intensifies with each new notification. When the user starts looking at the orb, eye contact sensors recognize this movement and the orb switches to a text mode that shows the number of messages and a summarized version of the message. The user can also interact with the orb by touching it. When the user looks back at his primary screen, the orb "reverts back to its idle state." An evaluation with several participants showed, that almost all of them preferred the orb towards the notifications of an email program, because the orb "allowed them to more quickly evaluate whether to accept or ignore the notification."

2.3.4 Light Systems

Matviienko et al. [21] analyzed over 70 ambient light systems and categorized them based on their purpose. Almost half of the systems display the status of something, followed by notification information or the current state of a process. Systems that display a status mainly use different colors to indicate something, while notification systems work with different brightness settings.

Ambilight[1] is a patented product used for televisions from the company Philips and was introduced in the year 2004. Small RGB-LEDs that are mounted behind a TV screen project a soft light onto the wall behind the TV, while the color is dynamically adjusted to match the current TV picture [33]. User studies from Begemann [30] found out, that Ambilight can reduce the visual fatigue by 60-90% of the people, while Plischke et al.

[1] Philips Ambilight http://www.philips.de/c-m-so/fernseher/p/ambilight

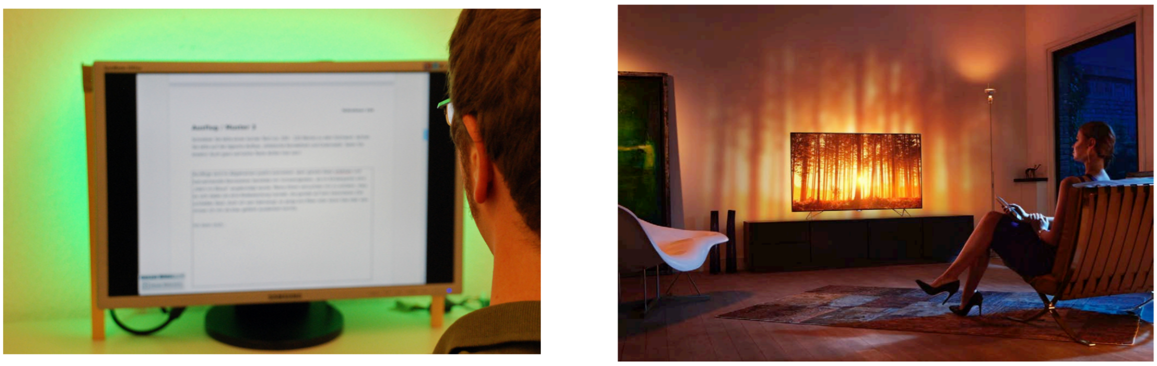


Figure 2.3: *Left:* Ambient Timer [24]. *Right:* Philips Ambilight [33].

[13] analyzed the hearth frequency of their participants and came to the conclusion that Ambilight reduces the brain's mental workload significantly.

AmbiLux is the newest invention from Philips and uses small micro projectors instead of LED lights. Thus, not only the colors of current TV content can be projected onto a wall but also movements and blurry textures. The similar project *IllumiRoom*[2] from Jones et al. [18] tries to augment the area around a TV screen with protections and visualizations based on the current content that is shown. The system is mainly build for video games and uses the *Kinect camera*[3] to scan the room geometry and furnitures. Based on these information, the system can then render the illusions and display them via a small wide field of view projector. The system supports physical interactions, for example elements can "break the screen barrier and fly into the physical environment" or the field of view of the game can be extended to all sides. The authors also thought that these ideas could be used for films, e.g. a grenade flies through the screen and virtually "explodes" in the living room.

Müller et al. [24] explored the idea of gaining the attention of a user towards a new notification by using an Ambilight-similar system. The wall behind the monitor is used as a surface, so that the user can perceive light changes in his peripheral vision. While the user is working on a task, the lights indicate a new notification by changing their color or their brightness over time. Based on a color pattern that ranges from green to red, the system can also be used as a timer in order to know when the current task should be finished to reach the next meeting. They called this concept *Ambient Notifications* and evaluated its effectiveness in a lab experiment. They found out that their ambient light can be used as a non-distracting reminder for the remaining time for a task and

[2]Microsoft IllumiRoom https://www.microsoft.com/en-us/research/project/illumiroom-peripheral-projected-illusions-for-interactive-experiences
[3]Kinect camera for Xbox One http://www.xbox.com/de-DE/xbox-one/accessories/kinect-for-xbox-one
[4]Images from Brett Jones http://brettrjones.com/illumiroom

Figure 2.4: IllumiRoom augments the area around a TV screen with protections based on the current screen content[4] [18].

that their system is equivalent to other reminding techniques like notification popups. Due to the fact that the produced light can be seen by everyone in the same room, the participants had different opinions on the system. Some saw the benefit that in an office environment everyone becomes more aware of the individual schedules. Others wanted to disable the system while working in a group or having a meeting with customers.

2.4 Smart Environments and Notification Systems

In 1998, Wisneski et al. [41] took various ambient devices and installed them in a small cabin, called the *ambientROOM*. "Putting the user inside the computer" instead of using a simple computer screen was the main idea behind this "augmented environment", that should improve the connection between a user and his environment. Figure 2.5 shows the ambientROOM with the supported different output channels light, sound, airflow and physical motion. Water ripples at the ceiling, light patches at a wall or the sound of "dry-erase pens rubbing against the board" are indicators for the activity of nearby persons but these cues can also be used to present stock values or network traffic.

In 2005, Helal et al.[15] developed a smart home system, where the residents could be informed about different states and activities inside the house. Sensors in the floor or mailbox can identify and locate people or notify about the arrival of new mails. A smart closet can suggest clothing based on the data for the outdoor weather and smart mirrors display important messages or reminders. The architecture of the system consists of different layers that connect the various devices and components with each other. Compared to other architectures, their whole system is context-aware so that state information can be abstracted and used for actions. For example, if inside the house a "hot temperature" (however hot is defined) is detected, the air conditioner can be turned on by the system. Additionally to the sensors, the people can interact

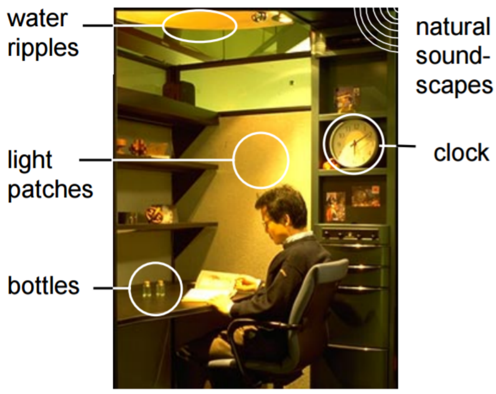


Figure 2.5: The ambientROOM contained various ambient devices in a small cabin [41].

with actuators which are physical devices. For example, if a lamp is turned on, a light sensor will detect the new state of the light. Sensors and actuators can be used to avoid inconsistent and undesired states, for example that the air conditioner and the heater work simultaneously.

Alt et al. [1] investigated the issue, to which extent public displays could be used to display content from individuals because they are often restricted to show only advertisements or content from the display owner. So called *public notice areas* (PNAs) are the traditional form of public displays and can be found in restaurants or stores. The notes there mostly rental requests, sales or services, such as babysitting. To make the digital posting of content as easy as possible, users can either use their smart phones or a website. They can also retrieve content on their phone by scanning a QR code next to each post.

Arlein et al. [3] proposed a framework architecture that uses different protocols like SMS, HTTP or SIP to generate, distribute and display notifications. Different architectural layers provide information about the current context of a user or a user profil. With all these data, each notification is adapted to the capabilities of an output device, for example if it can respond in real time or display a certain format, like text, voice or video. They also proposed the idea, that the current media stream on a device could be paused and replaced with a notification. Corno et al. [7] proposed a similar system as Arlein et al., but implemented machine learning algorithms to distribute notifications and learn behaviors from previous situations. Based on the awareness of the current context and the user habits, the system can decide who shall receive a notification in which situation on which device and in which output format (vibration, sound or light signal). Based on various information, like the current user activity and the user location, a so called *Decision maker* determines the output device as well as the output mode. A *Dispatcher module* adapts the notifications to the output capabilities of each device and sends them to all devices.

Vastenburg et al. [35] conducted a user study in a living-room laboratory with 20 participants in order to understand “how people experience notifications during everyday activities.” A set of messages was created with diverse urgency levels and corresponding texts, like *“Coffee is ready”* (low-urgency), *“Garbage will be collected tonight”* (medium-urgency) and *“Smoke has been detected in the shed”* (high-urgency). The notifications were projected to a wall in form of yellow post-it notes and two ambient color lamps to the left and right of the projection changed their color from orange to yellow to attract user’s attention too. Depending on the urgency-level, the whole setup was animated faster with different colors. The results showed, that the participants liked high urgent messages immediately and intrusively while the acceptance of the low- and medium urgent messages “could be improved by adapting the level of intrusiveness.”

2.4.1 User Localization

Previous work from Helal et al. [15], tried different ideas to locate a user in a closed environment. An “acoustic-based location system” was installed in the ceiling of each room and a transceiver send chirps into the environment. The user had to wore vests with integrated transceivers that “would listen for the chirp and respond with their own.” The discarded this idea because of the high costs and the intrusiveness of the vests that “defeats the desired transparency of a pervasive computing environment.” Instead, they used pressure sensors in the floor, which were installed and wired in each room to locate a user.

Since that time, the BLE technology emerged, which is is an extension of the Bluetooth standard and optimized for a low energy consumption in mobile devices[5]. Small devices like BLE beacons periodically broadcast short messages that can be used to detect the proximity to a specific location based on the RSS. Beacons are inexpensive, they can be used indoor or outdoor and provide a “greater precision than offered by alternative technologies, such as GPS, Wi-Fi and cell tower triangulation” [12].

Previous work showed that BLE beacons can successfully be used to locate precisely the location of person inside a building. Joonghong et al. [25] proposed a Trilateration-based algorithm, that works with BLE beacon signals and compared and evaluated it to similar algorithms in a real house. La Delfa and Catania [10] combined a BLE beacon system with a visual tag system to divide an indoor area into small regions and locate the user via a navigation app.

[5]Bluetooth Core Specification Version 4.2 https://www.bluetooth.com/specifications/adopted-specifications

3 Concept

This chapter explains the concept and idea of our notification system that displays user notifications in a home environment. First, we explain our general idea of a notification system and categorize possible output devices afterwards. We also give some detailed explanations, how notifications could be displayed on the output devices and show some visual examples.

3.1 Introduction

For our vision of a notification system we wanted to “detach” the notifications from a smart phone and display them in the nearby environment. Different output devices can be used and combined to display incoming notifications to replace the frequently checking of the phone. All output devices should be simple household appliances, and that the whole system can easily and ambient be integrated into every home. Some of the ambient information systems that were presented in the related work chapter are very “special” devices, that do not serve any other purpose than displaying information. In our opinion, such devices should have a further purpose if possible, like displaying the current time or that they could be used as a light source, so that they are not completely useless if there is no information to display. The user’s attention shall be directed towards a new notification, but he shall not be distracted from his current activity or task.

Figure 3.1 shows an example of an apartment with various devices that could be used to display notifications. Lights can notify in a simple and decent way or users listen to messages from speakers, which could otherwise not be read because of the current activity. Displays, like TVs, projectors or tablets can visualize the notification information in numerous variations and they can also be used to display images that were received through email or messaging notifications. Users can customize which information of their notifications shall be displayed and the data of each notification is analyzed by the system, so for example different priority settings can be distinguished and therefore the notifications are presented differently to the user. The system should be able to be used by multiple users at the same time and assign the incoming messages to each person, for example every notification is marked with a special color in order to distinguish

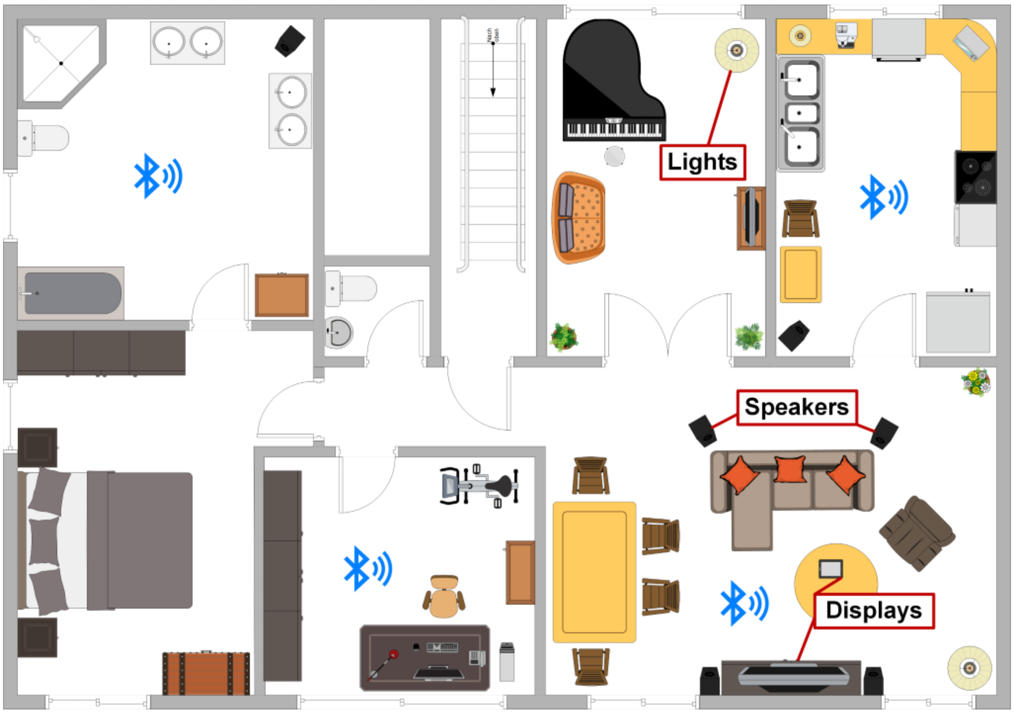


Figure 3.1: Various everyday objects can be used as an output device for notifications in a home environment.

them from others in a visualization. Instead of using all devices that are connected to the central system every time a new notification is posted, we only want to use the devices that are in the "near" of a user. Small BLE beacons and BLE scanning devices (see chapter 2.4.1) can localize each user and determine his current room. By this, we can save energy and prevent the unauthorized reading of personal messages if multiple persons are present.

We also wanted to give users the possibility to determine the behavior of the system in specific situations. We mentioned that for users it shall be possible to customize the information they want to display on the output devices. In a scenario, in which the user has visitors but still wants to be informed about new emails, he can specify that he wants to see emails on a nearby display but without any text content. If no voice or sound output is desired during a conversation or TV viewing, the users can control this behavior by simply changing the ringer mode on their phone. Android phones make use of three different settings for the ringer mode to notify the user: the *normal mode* (sound and vibration), the *vibration-only mode* and the *silent mode* (neither sound nor vibration). According to Pielot et al. [26] "people frequently disable sound, but rarely disable all alerts." Using the silent or vibration mode on the phone is recognized by the system and causes, that only visualizations or lights are used to notify the user. Changing the setting back to the normal mode reactivates the voice output.

3.2 Output Devices and Methods

This section will list and rate different devices and methods that could be used in a home environment to display notifications. We will also describe suggestions, how notifications could be adapted for each device type based on its capabilities.

Figure 3.2 shows a possible classification of the device categories with regard to their level information content and their level of intrusiveness. Display devices can display many information simultaneously combined with a low level of intrusiveness. Smart phones and tablets can use small vibration motors as an additional output channel. Voice output can provide a similar level of information, but in some situations it can be annoying and inappropriate. Although light systems can barely display the contents of a notification, they are perceived less annoying compared to the other devices.

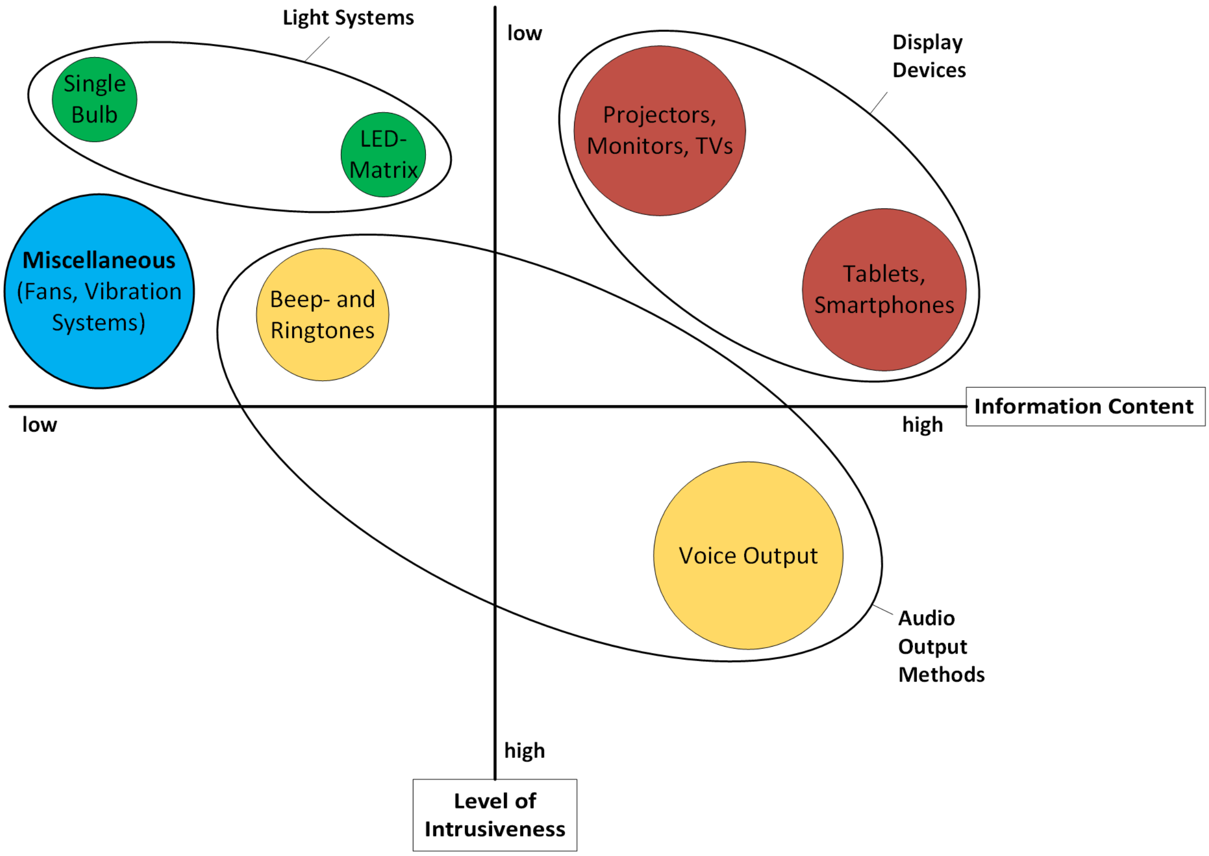


Figure 3.2: Categorization of the different device types according to their level of intrusiveness and level of content.

3.2.1 Audio Output Methods

Sound and audio effects as well as voice output can be a great and powerful tool when it comes to the presentation of information. The parameters *volume*, *pitch* and *sound color* offer a wide range of options and even more when each parameter changes within a time interval (for example a steadily increasing loudness, called crescendo in music) while a computer voice can read the whole message text. Users can be notified immediately based on the chosen parameters and voice output offers the possibility to listen to a notification directly. But as great as these advantages are, they unfortunately stand against the disadvantage that there can be only one output at any time. Thus, several messages would have to be grouped together or queued in order to process them one after another. This restriction could be neutralized by several speakers, but if multiple notifications are read at the same time, it would be difficult to listen to each of them. Furthermore, sound and voice messages can be perceived as very annoying and inappropriate in some situations, if they happen for example during a conversation.

Through to the fact, that the different information of a notification can not be played simultaneously, the following output sequence could be used.

Priority
By defining three different beep tones, messages could be divided into the categories less important, important, and very important. This allows users to decide whether to listen to the following or not when hearing the beep tone. A similar effect can also be observed in the radio: before the announcement of the traffic news one can here a specific beep tone, so that everyone knows what comes next and decide if he wants to listen or not. Alternatively, only one beep tone is used, which differs in its pitch or volume to cover the three categories.

Posting-App
The user's attention can be easily excited by using known sound effects or jingles (WhatsApp or email sounds). Thus, the corresponding app or the category of the notification (system, alarm, error, call etc.) can be identified. It would also be possible to encode the priority of the notification within the playback of the jingle, e.g. the WhatsApp sound is played louder or higher for the corresponding priority, but then every user must have set this information beforehand via an app or another user interface.

Content
An electronic voice could read the notification content, like the title, the text and the number of messages. This output method could also be adapted by the user to his current situation, so when someone is not alone, he could be notified about a new email, but it's content is not read by the system.

3.2.2 Light Systems

Compared to the other device types, light systems are probably the easiest to integrate into a home environment. They can be used in different sizes and variable quantity as well as different shining colors. Modern light bulbs like *Philips Hue* or *LIFX* can be controlled via a smart phone application so that the user can decide which *color* and *brightness* he wants to see in a certain situation. Like audio systems, it is possible to notify a user immediately based on the chosen parameters.

One great disadvantage of light systems is, that are very restricted when it comes to the displaying of information. They are mainly used to display a status, a simple notification information or the current state of a process [21]. They can not display an image of the sender or text parts of a notification. They can only notify the user, that he received a new notification from app X. A simple extension could be, that the light system blinks according to the number of messages. Another disadvantage is, that color changes are sometimes not visible because it is too bright in the room or there is just an unfavorable light irradiation of the sun.

The following will illustrate some ideas, how the different notification information could be displayed by light systems.

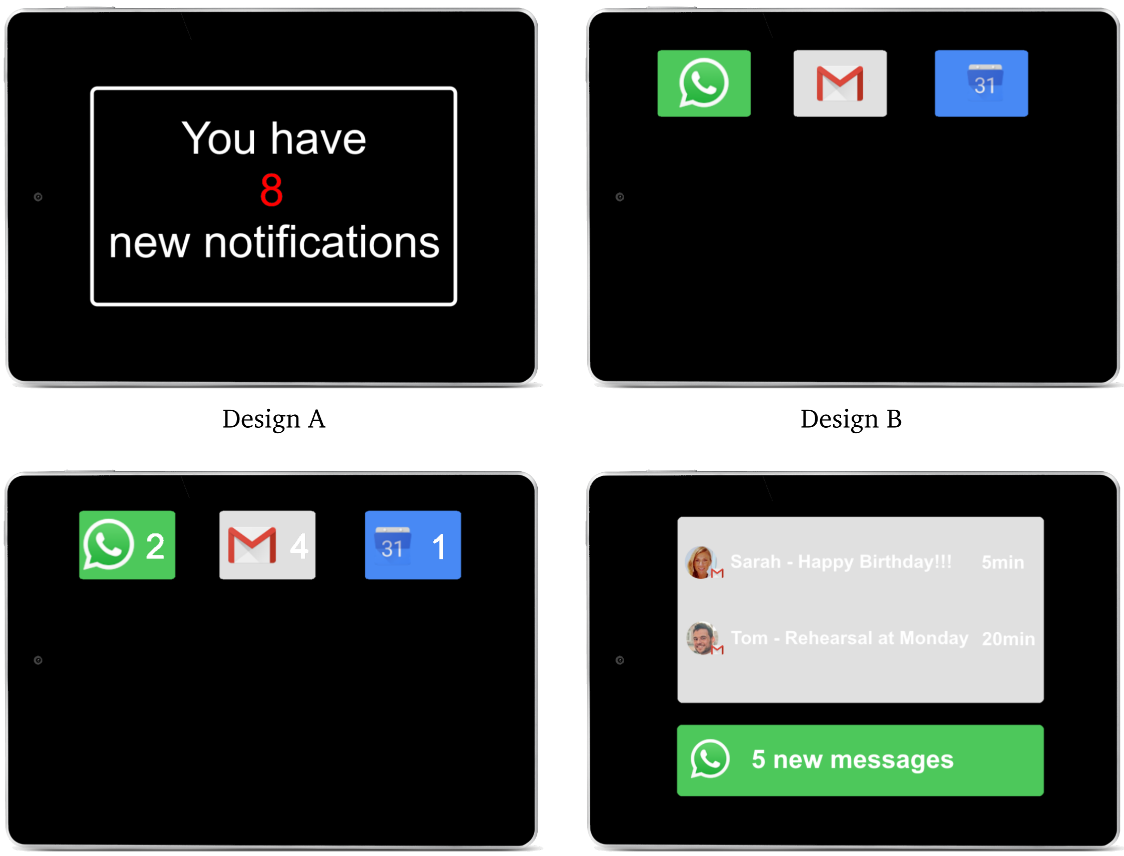
Priority
Slow or fast pulsing of the current color gives a hint if the notification is important or not. Instead, it would also be possible to change the brightness from a dark to a bright color or vice versa.

Posting-App
The color of the light bulb could correspond to the app icon of the posted notification, e.g. WhatsApp leads to a green color, Twitter to blue and so on. But if multiple app icons use the same color scheme, it is very difficult to distinct them from each other. If certain colors, like red, are linked with a warning, one would not expect that it could be a notification from an app, where red is the dominating color in the app icon (for example GMail).

Content
As already described, light systems can not display images or text. However, it would be possible to develop some kind of Morse code, which would have to be learned by each user or looked up in a list. The flashing of different colors could then be used to indicate the sender of a message or if the message contains a picture. (Red, Red, Green) might mean something different than (Green, Green, Red).

3.2.3 Display Devices

Display devices are the devices with the most output options. They can display a lot of different information at the same time and if supplemented with additional output channels, such as speakers or vibration motors, even more. However, too much information can lead to confusion and the excessive use of animations to an increased level of intrusiveness.



Design A | Design B | Design C | Design D

Figure 3.3: Output designs with an increasing level of information content[1].

For the presentation of the notification information, we have designed several output designs which can be considered in Figure 3.3 and Figure 3.4. Because ambient information systems shall not distract users, we tried to avoid and minimize any animations. However, an empirical study from Plaue and Stasko [27] found out, that if visualizations make use of simple and slow animations, like panning and zooming effects, they appear

[1]Images from uiFaces http://uifaces.com and FreeImages http://de.freeimages.com

Design E   Design F   Design G

Figure 3.4: Output designs with an extended and alternative look[2].

more interesting and motivating for users. Other experiments from McCrickard et al. [22] showed similar results with the note, "that the use of animation can assist in maintaining awareness without causing undue distraction in particular situations. As with most tools, peripheral animations can be and have been misused, but when used properly, this research has shown they have the potential to be beneficial."

The four designs in Figure 3.3 provide an increasing level of information content. Design A can notify a user that he received a certain number of new notifications, but he doesn't know from whom or with which content. Design B and C offer the next level of information and show which specific app posted a new notification (and a concrete number in design C), but the information about the sender or the content of the notification is missing here too. The highest possible information content is provided by design D. Both the sender name and the subject line are displayed, as well as the posting time of the message. If the sender has set a picture of himself in the posting app, this can also used and displayed.

Figure 3.4 shows, how additional information could be integrated into each output design. Design E can notify the user about important messages, like incoming calls, which can be identified by their priority field (see chapter 4.3.2 for more information). A red box is put over the existing design, so that the attention of the user is immediately directed to the display. Design F shows, how images that are nested inside a notification, could be displayed. Design G shows an idea, how the notifications of multiple users could be displayed. Each user is assigned his personal color so that he can identify his messages on the screen. The size of a bubble also indicates the number of messages.

[2]Images from uiFaces http://uifaces.com and FreeImages http://de.freeimages.com

3.2.4 Miscellaneous

Devices in this category are not considered for our system, but for the sake of completeness they are mentioned here. Some ambient information systems, like the Ambient Fixtures from Dahley et al. [9] or the Dangling String from Weiser and Brown [40] belong to this category. Other devices could be simple fans, robot arms or vibration motors, actually every hardware that causes motion. They all provide an interesting and versatile way to inform the user about a new event or message, but similar to light systems, their information content is very limited and it could be difficult to integrate them in a home environment.

4 Implementation of the System

This chapter will describe the implementation of our notification system in four different sections: the first section shows an overview about the system's architecture. The second section describes the existing implementation of the meSchup platform that we used for our implementation. The third section contains the extensions that were added to the existing Android app. Finally, we explain the notification pipeline and the concrete implementation of the user localization that is realized on the server side.

4.1 Architecture

Our notification system has been developed on the basis of the *meSchup* platform [19] that provides tools to easily connect different devices with a server platform to configure and control them. Figure 4.1 shows our architecture in a simplified version.

For our approach we used the existing meSchup Android app and extended it, so that it can listen for user notifications and send them to a central server. The server receives the notifications and processes them in our so called *notification pipeline*. The pipeline uses information about user-device relations as well as the current room of a user. Small BLE beacons that are carried by each user, periodically broadcast their signal strength between them and a BLE signal receiver to the server. Based on configured thresholds, the server decides if a user is in a specific room or not. Instead of using all devices that are connected to a server, we use the information about the current user room to "route" notifications only to devices that are in the user's proximity. The final step of our pipeline uses several created output renderer that adapt the notification information to the capabilities of each devices. On the output side, we use tablets that run the Android platform, Raspberry PI systems that can be connected to external monitors or projectors and LIFX bulbs.

Our system is similar to the systems from Corno et al. [7] and Arlein et al. [3]. We also use text, voice and lights as output mechanics, together with information about the current room of the user. But our pipeline does not consider the current context of a user and if multiple output devices are available, we don't choose "the best" device for a specific situation. Instead, we focused more on the visualization part of our system.

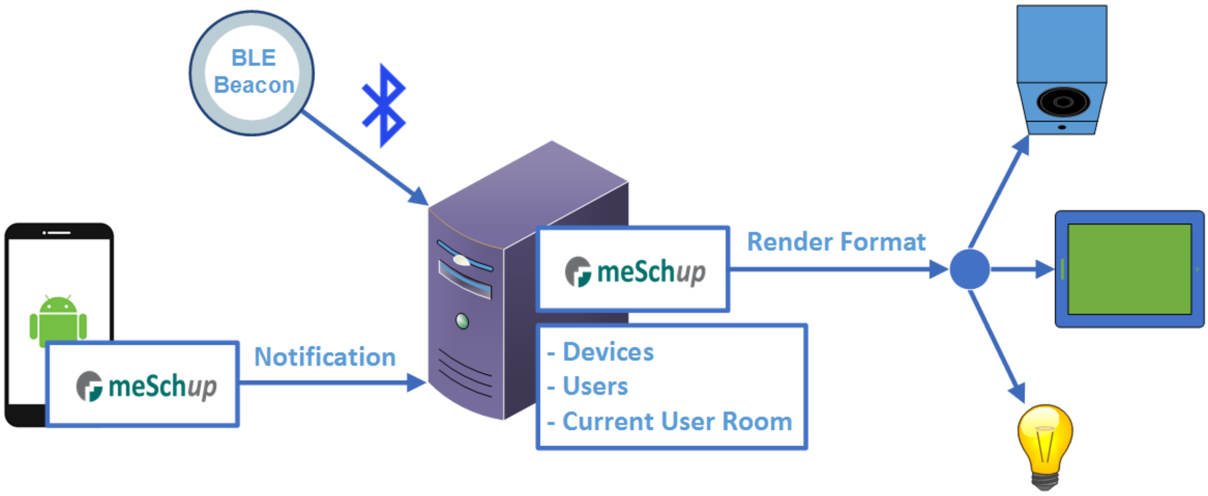


Figure 4.1: Architecture of our notification system.

Visual output designs that present the notifications to the user can easily and fast created through to the fact that they are simple websites based on HTML, CSS and JavaScript files. Users can also perceive, if they are "connected" to a display device, since we display small circles in the lower left corner of each design which present a user. Our system is also capable to work with multiple users. Each user can select his personal color via the Android app, which is then used every time to display his notifications and distinguish them from the notifications of other persons.

4.2 Existing Implementation

The existing implementation of the meSchup platform consists of two different parts that were used and extended for this thesis: a server part and a client part, that is available for Android devices and Linux based systems.

The server platform is used to connect different devices to the platform and configure them with so called *Modules*. Modules are available for each device based on its type and provide different functionalities. For example, if a device supports the *Audio module*, it could be configured with it to play sound and audio files or the *WebDisplay module* loads and displays web content from HTML pages. All connected devices and their behavior can be controlled over so called *Interaction Scripts*, which are written in JavaScript. The devices and their modules can be referenced within a script to call their functions. For example, if a device has been configured with the WebDisplay module, we can call the module's *showURL* function, send a URL of a web site to the device and the module loads the site to display it. After a device has been configured with a module on the server side, the configuration is sent to the device encoded in the JavaScript Object Notation (JSON). The client software on the device receives the configuration, decodes it and loads the corresponding module on its side.

The communication between a device and the server uses UDP datagrams to transmit the JSON objects. However, not only device configurations or function calls can be sent to the devices, the devices can also send data back that are interpreted as so called *Events*. Events are incoming device messages on the server side, which can contain different data. They are listed in a table on the server platform, so that the user can monitor them and use the Event data in the Interaction Scripts. A device with a configured *BLE module* can scan its vicinity about other BLE devices. If a Bluetooth signal from another device has been detected, the module on the client side creates a new JSON object that contains the MAC address of the found device and the current signal strength. The object is sent to the server and received there as a new Event. Chapter 4.4.1 explains, how we use this mechanic to identify the current room of a user. Another type of Event is sent by the Accelerometer module, that is available for Android devices. This module *"measures the acceleration force in m/s2 that is applied to a device on all three physical axes (x, y, and z)"*[1] and sends it as a new Event to the server.

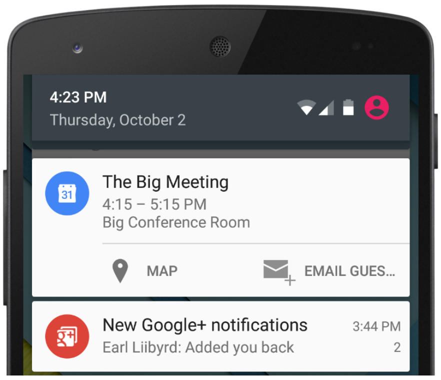
4.3 Android App

In this section, we will first show the presentation of notifications in different systems. After that we describe the implemented changes that were made to the meSchup Android app. We added a new module that is loaded as an Android activity, to receive and process notifications that are on the user's phone. Via a graphical interface, the user can control which notifications from which apps shall be sent to the server as an Event, so that the notification data can be used in the scripts on the server side.

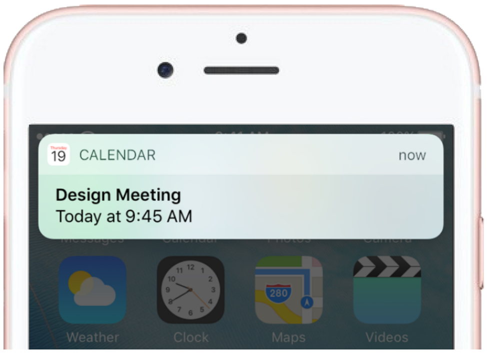
4.3.1 Notifications in different Systems

Notifications can be found in many systems, including the most popular platforms from Google, Apple and Microsoft. Figure 4.2 shows different notification implementations for mobile and desktop operating systems. They have in common, that the notification is always displayed inside a rectangular, containing an icon at the left side and a title and text field. These parts are used to identify the app or program that created the notification as well as the sender (if it is an email notification) or the upcoming event

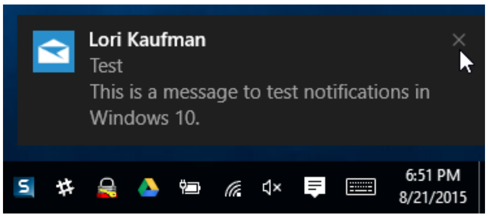
[1] https://developer.android.com/guide/topics/sensors/sensors_overview.html



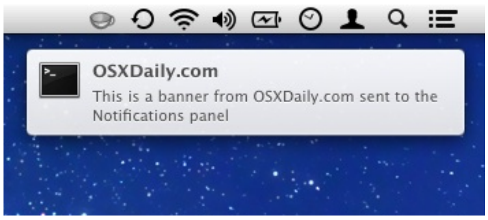
Google Android 5.0[2].



Apple iOS 10[3].



Microsoft Windows 10[4].



Apple OS X[5].

Figure 4.2: Notifications in different systems.

(calendar notification). The mobile systems display the notifications at the top border of the screen, while Windows uses the lower right and OS X the top right corner. They can be dismissed or closed by swiping to the right side, by clicking a small "X" or they disappear after a couple of seconds. Depending on the system, notifications can be accompanied by a simple sound or vibration pattern.

4.3.2 Notifications in Android

Since Android 4.3, every notification, that is posted by an app, is represented as a *StatusBarNotification* object within the Android system. Each of this objects consists of different fields like the post time or the package name of the app that posted it. For a better manageability of the notifications and to gain the option to add or remove different information fields from a notification, we don't use these objects themselves. Instead, we

[2]Google Android Developer API Guides https://developer.android.com/guide/topics/ui/notifiers/notifications.html
[3]iOS Human Interface Guidelines https://developer.apple.com/ios/human-interface-guidelines/features/notifications/
[4]Image from http://www.howtogeek.com
[5]Image from http://cdn.osxdaily.com/

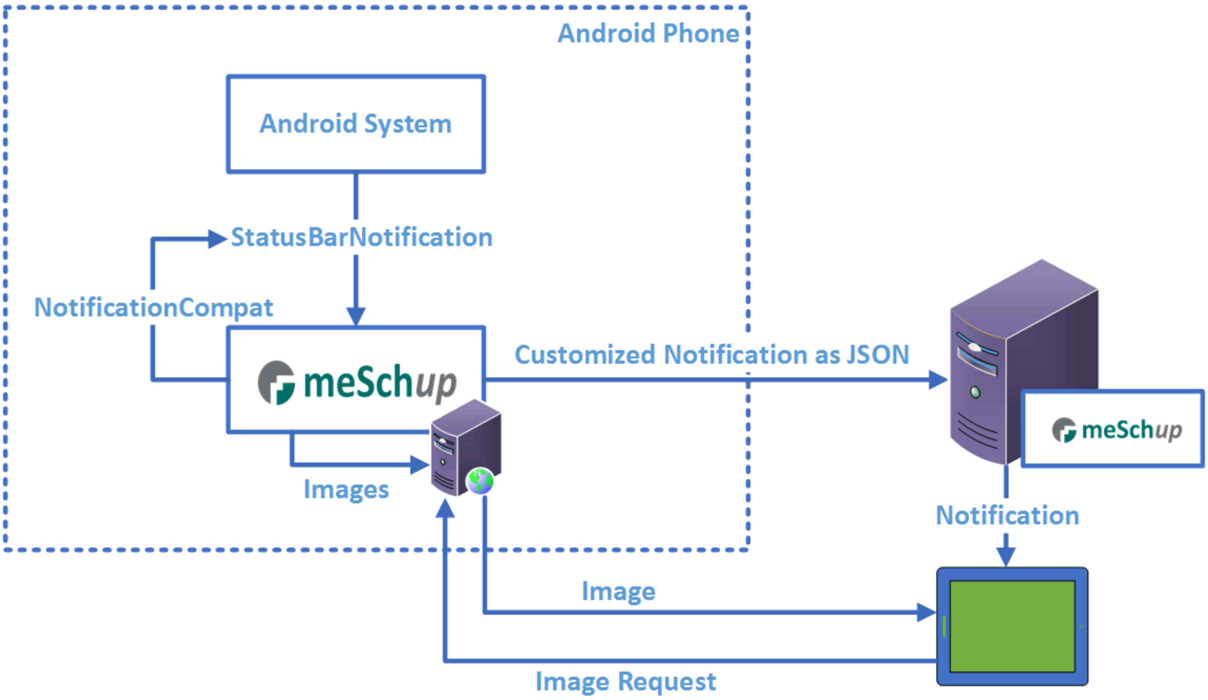


Figure 4.3: The left part shows the process for creating a customized notification. A web server that runs in the background of the app is used to deliver images that are requested by other devices.

build our “own”notification object in our Android app, so that we have the full control which information is sent to the server. Unfortunately, every StatusBarNotification object supports only a small number of data fields. To access additional fields, we use a helper object provided by Android, called *NotificationCompat*. With this helper object, we can access fields, like the title or the text that is shown to the user or even images that are inserted in the notification. Some of the most important fields can be seen in Table 4.1 (for a full list of all notification fields see appendix A.1). The left part in Figure 4.3 shows the process which generates such a customized notification. Every time a new notification has been posted by an app or dismissed by the user, we can access the StatusBarNotification object through the Android system and build our customized notification object based on the settings the user selected (see chapter 4.3.6). After the object has been created, it is sent as a JSON object to the server. Listing 4.1 shows an example for the result of this process.

Some apps use the option to stack multiple notifications, so that they are displayed as one notification to the user. These *summary or group notifications* are often created by email or messaging apps, but it depends on the implementation of the app that posted them. However, the *Facebook Messenger* app is an exception here, because it only groups the notifications from one, but not from several people and instead creates always a

Field name	Data type	Description
appName	String	The name of the app that posted this notification.
packageName	String	The package name of the app that posted this notification.
postTime	long	The time in milliseconds when the notification was posted.
category	String	Categories can be used to describe the content of a notification. Examples for this field: ALARM, CALL, EMAIL, ERROR, EVENT, MESSAGE, SOCIAL, SYSTEM.
priority	integer	The priority of a notification can be used as an indicator, how important the content of this notification could be for the user.
key	String	A unique key for notification that can be used as an identifier.
title	String	The title of the notification.
text	String	The main text line.
textLines	String	An array of multiple text lines. Each line ends with a new line character.
ringerMode	String	Contains the current setting of the *ringer mode*, possible settings: normal, vibrate, silent.
doNotDisturbMode	String	Contains the current setting of the *do not disturb mode*, possible settings: disabled, important, silence, alarms.

Table 4.1: Descriptions for some of the most used fields in our notification objects.

new group notification. Another example for a deviant behavior could be observed with *Skype* notifications: this app always generated two notifications with the same content. We had to filter one of them, to not confuse the user. For our implementation of a notification system, we had to consider and test all these different settings to provide a consistent user experience.

Listing 4.1 Example for a customized notification object encoded in JSON.


```
{
      "ringerMode": "vibrate",
      "doNotDisturbMode": "disabled",
      "notification": {
            "status": "new",
            "appName": "WhatsApp",
            "packageName": "com.whatsapp",
            "postTime": 1476989863106,
            "key": 1|com.whatsapp|10255,
            "priority": 0,
            "title": "WhatsApp",
            "text": "4 messages from 2
                conversations",
            "textLines": "John: How are you
                today?\nMary: I heard you
                broke your leg, are you ok?"
      }
}
```


Posted notification.


```
{
      "ringerMode": "vibrate",
      "doNotDisturbMode": "disabled",
      "notification": {
            "status": "removed",
            "appName": "WhatsApp",
            "packageName": "com.whatsapp",
            "postTime": 1476989864156,
            "key": 1|com.whatsapp|10255,
            "priority": 0
      }
}
```


Dismissed notification.

4.3.3 Notifications in Android 7.0

While this thesis was written, Google released the new Android version 7.0, that brought some enhancements for notifications. The visual design of notifications lightly changed and by swiping one of them only a little bit to the right side of the screen, the user can now decide, if notifications from that app shall be blocked or displayed without a beep tone. It is now also possible, to directly reply to a text message within the notification interface. This option was partly implemented from some apps like WhatsApp, but now it is fully supported by the Android system [11, 6].

4.3.4 Managing of Notifications

When the user configures our module the first time and starts the meSchup app, he is asked to grant access to the notifications on his device (left picture in figure 4.4). For security reasons, the notification access is disabled by default for each application by the Android platform. The user must handle the access permissions individually for each app that asks for it. To do that, the Android system settings provide a separate menu, in which the different apps and their current permission status can be seen and changed (right picture in 4.4). After the user granted the meSchup app the required permission, it is possible to receive every notification that is generated on the device and handle it with our app.

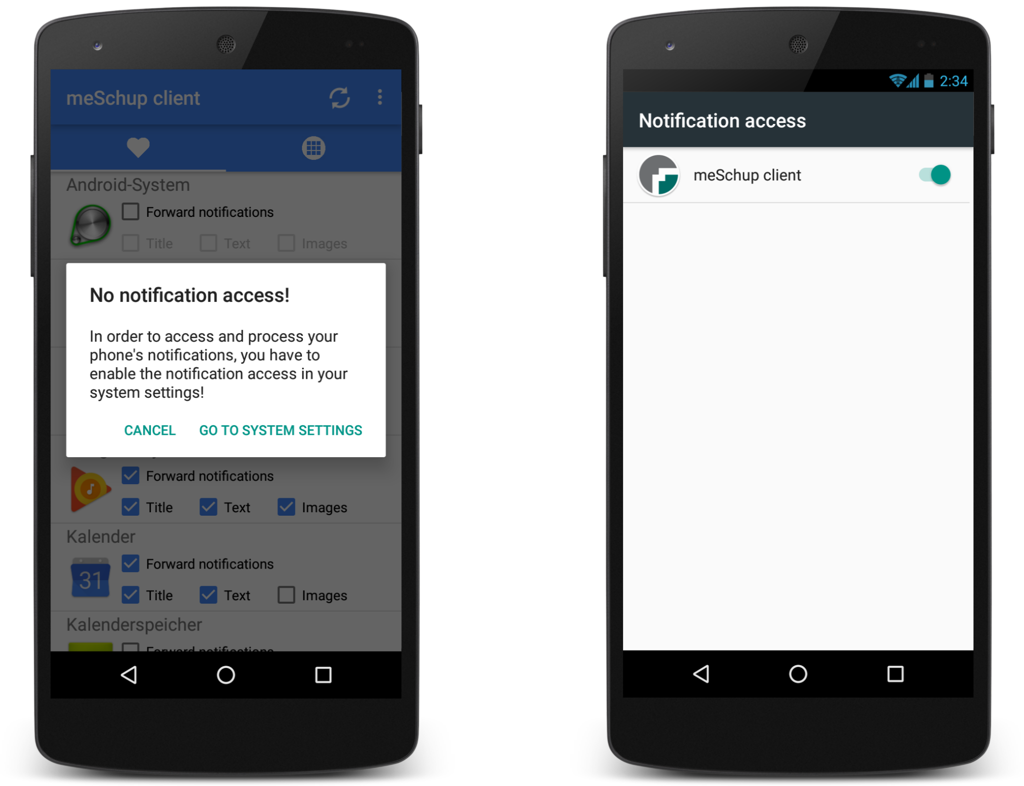


Figure 4.4: The notification access for an app must be set manually by the user.

4.3.5 Storing of Images

Notifications are not restricted to contain only text elements. Notifications from messaging apps like WhatsApp can also contain images that were sent between two users. Figure 4.5 shows a notification in Android with an integrated image. All in all, we have three images in this notification: the sender's profile icon, which is set in the app that posted the notification, the app icon at the right bottom corner of the profile icon and an image that was attached by the sender of the message.

These images are an important factor when it comes to the presentation of the notification on output devices with a display. The users can recognize with just a quick glance on the display, which app posted the notification and who was the sender. Furthermore, the image inside the notification can be displayed much larger on a tablet computer or a big TV screen compared to the display of a smart phone.

To display the images, we needed a central place that could be used to store the images so that every device that is connected to the system can access them. Each image in a notification is encoded in the Bitmap format and we can use the NotificationCompat object again to access them. In our first attempt to distribute the images, we tried to

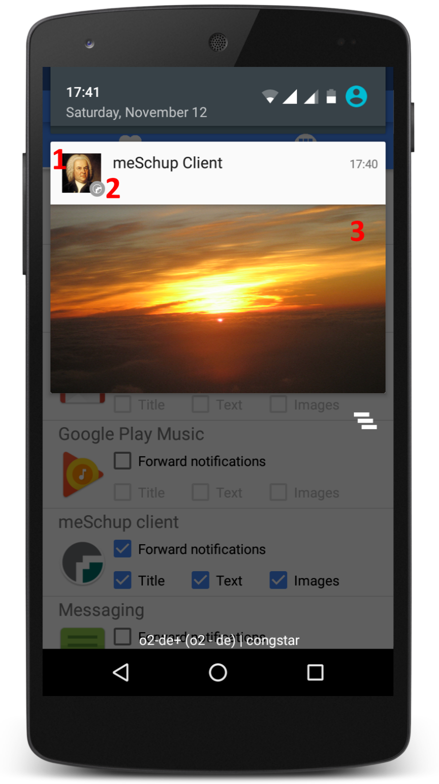


Figure 4.5: Example for the different images a notification can contain, 1) profile icon of the sender 2) app icon 3) attached image from the sender, embedded in the notification.

take the Bitmap format, encoded it to a Base64 string and sent this string with the rest of the notification to the server packed in a JSON object. We discarded this idea very quick, because we had to drastically limit the size and quality of each image, so that it was even possible to sent it within a UDP datagram.

In our second and finally successful attempt, we stored the images on a web server that runs in the background of the meSchup app, so that it is possible to access the images, even if the app is closed. Each of our customized notifications contains now only the URL to the image so that a device can request it from the central location. For the web server, we used a small Java library called *NanoHTTPD*[6], which starts the server each time when our new module is configured on the server side. After the module has been disabled for the current phone, the server is shutdown. If the user dismissed a notification on his phone, the notification images are deleted from the server too. Figure 4.3 shows this process on the right together with the web server on the Android phone.

[6]https://github.com/NanoHttpd/nanohttpd

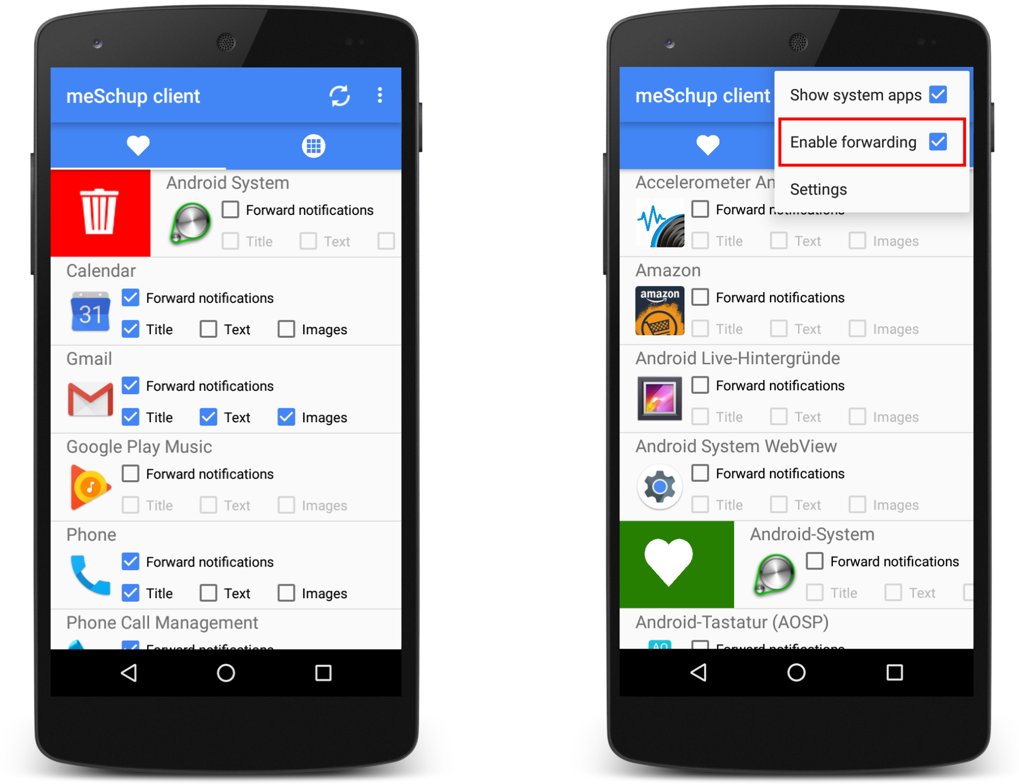


Figure 4.6: Via different checkboxes the user can select which information of a notification shall be forwarded to the server.

4.3.6 Customization of the Notification Data

For the user interface of our app, we tried to integrate different options, so that the user can decide at any time which notifications from which app shall be forwarded to the server and which data fields it shall contain or not. Figure 4.6 shows the final interface in our Android activity. Via the tabs at the top of the screen, the user can access two lists, filled with different apps. The right list contains all apps that are currently installed on the device. Apps that are listed here can be added to the left list by swiping to the right side of the screen. The idea behind the two lists is, that the user can fill the left list with his “favorite” apps or apps that are used frequently, so that a quick access is possible.

Each app entry contains four checkboxes, that gives the user the control over specific parts of the notification. If the top checkbox is checked, each notification posted by this app is forwarded to the server. With the checkboxes below, the user can select if a forwarded notification shall contain specific information like the title, the text or even images.

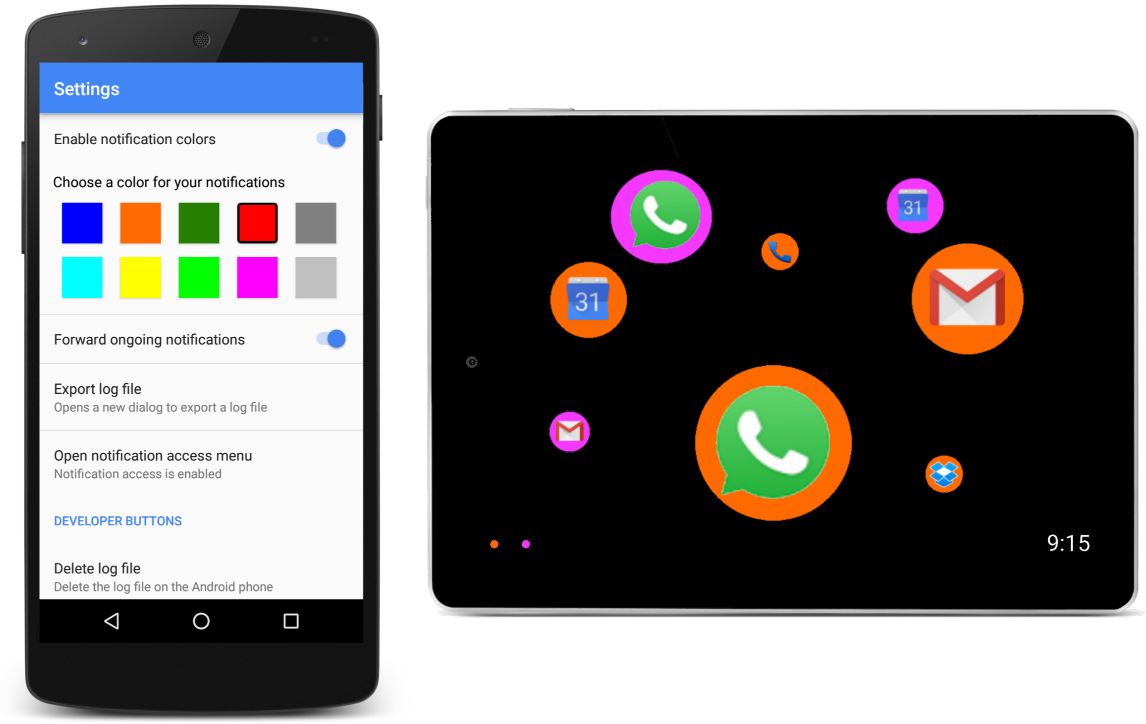


Figure 4.7: Settings screen of the app that can be used to choose a color for the notifications. The right picture shows an multi-user example, where the notifications of each user are presented in a different color.

The favorite app list combined with the four checkboxes allows the user to react to different situations in a fast and simple way: if the user sits in front of a TV and wants to watch undisturbed a movie, he can disable the forwarding of all notifications, by using the checkbox in the top right corner of the user interface, or only for some specific apps. After he finished the movie, the forwarding option can be enabled again. In another example, we can think about situations that involve multiple persons in the same room. Our user wants to be informed about notifications, but in a discreet way. So he can select, that his notifications shall be forwarded and displayed on a device, but without any specific information, like the sender's name or any text lines.

To support multi-user setups, each user can choose his personal color via the settings screen, so that the border or background of his notifications is displayed with his selected color, seen in Figure 4.7. The colors are disabled by default, if no color is selected the default color is white. After the user enabled the colors and selected one of them, the new color is send as a hex value within a JSON object to the server. The server can now forward the color value to each output device that has a display, where the whole design is refreshed with the new color.

4.4 Server Architecture

To receive the notifications from the Android app as a new Event, we developed a new module that can be configured and added to any registered Android device. Therefore, the incoming notifications can be processed in our notification pipeline and rendered for the connected output devices.

4.4.1 User Localization

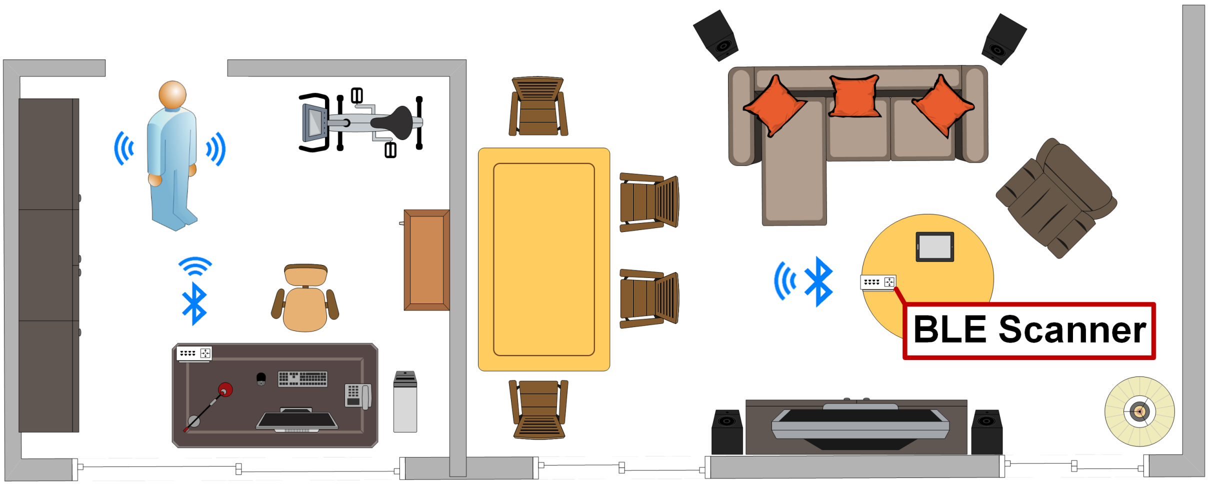


Figure 4.8: The current position of a user can be located via personal BLE beacons and BLE scanners.

Every time a new notification is received by the server, it is sent towards the output devices by our server scripts. Without further additions, the scripts would always use all devices that are registered to the platform. But in many situations, this isn't the best idea. When the system is used by several users, each of them could easily see, read or listen to the notifications of the others. Most of them would not want to use the system furthermore after another person had read their emails or WhatsApp messages. To avoid such rejection, we had to consider some improvements.

Fortunately, the meSchup platform allows us to assign each device to a location by filling out the appropriate field in the configuration screen of a device. With the setting of the field, we can define which devices are in which room and route the notifications in the "near" of each user. Thus, we have to control only some of the devices if a new notification comes in and we can also prevent the unauthorized reading.

Now, how can we find out where a user is and which devices surround him? In our former Concept chapter, we said that we use the Bluetooth technology to identify where

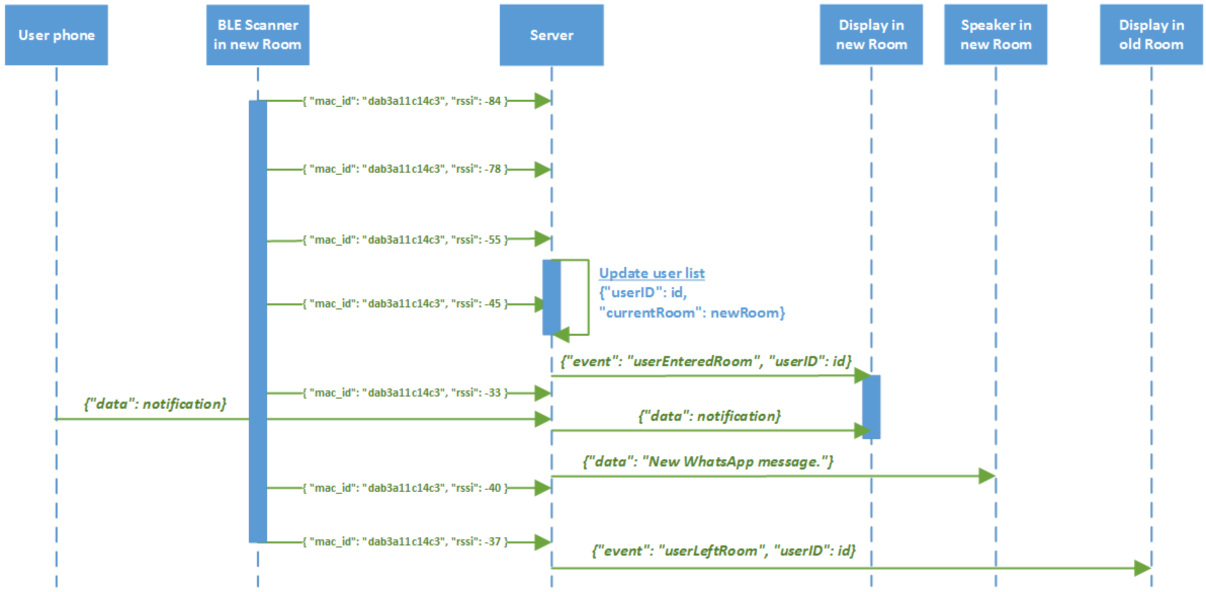


Figure 4.9: Sequence diagram for the room detection of a user.

a user is. Every user has an assigned BLE beacon, which is a small wireless device that periodically broadcasts a BLE signal. The signal can be received by a BLE scanner to determine the position of the beacon and therefore of the user. For the BLE scanner we use a simple Raspberry Pi to scan its environment via a Bluetooth USB dongle. For every BLE beacon that is detected, it sends a JSON object as a new event to the server, that contains the mac address and the received signal strength.

To illustrate this process, we took a section from our apartment graphics and added the necessary details. Each of this two rooms in Figure 4.8 is equipped with a BLE scanner that scans its environment continuously. If a user comes in the range of a scanner, he is detected by his personal BLE beacon and the scanner sends its events to the server. To ensure that not every time something is triggered as soon as a new BLE event comes in, we defined virtual boundaries based on the signal strength. For example, if the signal strength is below a value of 70, we can assume that the user is in the same room as the scanner that sent the event. By filling out the location property for the scanner, we can get its room name and save it as the current room for the user in our global user list. A background task ensures, that a user is set to the *out of range state*, if a BLE scanner should not receive a signal from his beacon for a while. When a user enters a new room, all notifications that are active on his phone are sent to every display device in that room. If a user leaves a room, his notifications are removed from the displays, while the notifications from other users stay. With this procedure we can uphold a consistent user experience, so that the notification state from a phone is always represented on all displays. Figure 4.9 shows the whole process as an UML sequence diagram.

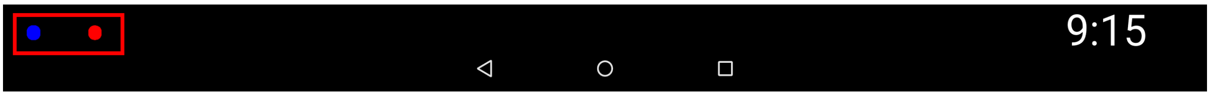


Figure 4.10: Small circles in the lower left corner of each output design give the users a simple hint, if they are connect with the display.

For display devices we added a bar at the bottom of each output design, which can be seen in Figure 4.10. In the right corner the user can see the current time, so that the display shows something useful even when there are no notifications. The small circles in the left corner represent each user that is currently in the same room as the device. Thus, the user gets some feedback if he's connected to the device or not. The drawing or removing of the circles is triggered by the *userEnteredRoom* and *userLeftRoom* in Figure 4.9.

4.4.2 Processing of Notifications

In this section, we want to explain the different steps that are executed once a notification has been received by the server as a new event. We implemented a pipeline to process the different information about the notification itself and the devices that are connected to the system. Each pipeline step is responsible to receive different information from the platform and filter them based on the available data. Figure 4.11 illustrates the pipeline and the following list explains each step in detail.

1. **Storage of Notifications**
   Each notification is sent as a JSON object, which can be accessed and modified through our scripts. In our first pipeline step we make sure, that all notifications are stored at a central place, so that each script can access them in a fast and easy way. The notifications are not stored in an array but a simple JSON object, which is used like a hash map: the key field of a notification is used as a unique hash map key, where the notification itself is the corresponding value. By this, we can access each notification directly over a key and don't have to iterate over an array or list each time we want to access an object.

2. **Retrieve the Device List of the User's current Room**
   A great advantage of our system is, that it is possible to locate the current room of any user who is registered to the platform. Therefore, only the devices that are in the near of the user are used to display the notifications. With the following steps, we explain the process that retrieves these devices in detail.

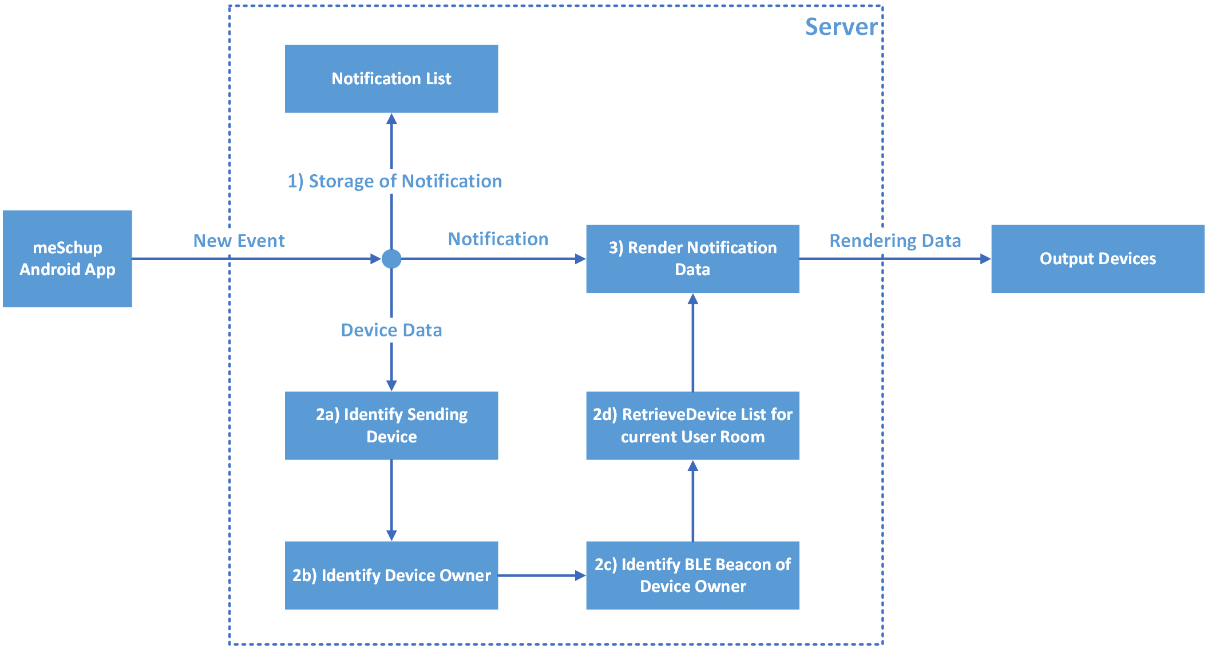


Figure 4.11: Notification pipeline that is realized on the server side.

a) **Identify the Sending Device**
After a new notification has been saved in our hash map, we can identify the device from which the notification was received. For that, we can use functions that are provided by the meSchup platform. In meSchup it is possible to identify the id of the device that triggered the last event and access its configuration. The device itself is stored as a JSON object with numerous data fields.

b) **Identify the Device Owner**
With the identification of the sending device in step 2a), we can now easily access its data fields and find out to whom the device belongs. Each device that is registered to the platform can be assigned to a person (owner) and to a location (room). These fields are normally optional, but for our system they are a mandatory requirement. If a device has no owner, we abort our pipeline here, otherwise it would not be possible proceed with the subsequent steps.

c) **Get the BLE Beacon of the Device Owner**
Now that we have the owner's name of the device that triggered the pipeline process, we can find out which BLE beacon is assigned to him. With the id of the beacon it is possible to get the entry of the owner from our user list, which can be used to get the current room of the user. As seen in listing 4.2, we iterate over the list of all devices that are registered for the platform, select only devices that are a BLE beacon and compare the name of the owner of this beacon with the name we found out in step b).

Listing 4.2 Step 2c) of the pipeline process that gets the JSON object of the BLE beacon of a user.

```
var userBeaconID;
for (var device in deviceList){
      if(device.type === "ble.beacon"){
            if(device.owner == ownerOfTriggeredDevice){
                  userBeaconID = device.uuid;
            }
      }
}
```

d) Get the Device List
With the id of the user's beacon, we can now get his entry from the user list and therefore his current room. With the room name, we can again iterate over the list of all devices again and compare the location of each device with the room name of the user. If both are the same, we put them in a list that can finally be used to display the notifications. As an extension, we can add devices with no location property. These devices can be used anytime a new notification comes in, independently from the user's current room.

Device type	Description
Displays	The output designs that are rendered by display devices consist of simple HTML, CSS and JavaScript files. To display the notifications on them, we call a design via the *sendURL* command of the WebDisplay module. Once a design has been loaded, we can send the JSON objects for each notification to it, so that we don't have to reload the whole page every time.
Speakers	A URL to an audio file is sent to the device, which is then loaded by the Audio module. If the Text2Speech module is used, we sent a String with different text elements to the device, e.g. *"New message from "+notification.title+":"+notification.text;*.
LIFX bulbs	A JSON object with the fields *hue*, *saturation*, *brightness* and *kelvin* is sent to the bulb every time a color change shall happen. An example could look like the following: {*"hue": 319, "sat": 100, "brightness": 100, "kelvin":0*}.

Table 4.2: Descriptions how the notification data is rendered for the different device types.

Listing 4.3 Step 3) of the pipeline process that renders the notification data for the different output devices.

```
for(var i=0; i<deviceList.length; i++){
      var currentDevice = deviceList[i];
      if(currentDevice.type === "android.base.web.view"){
            currentDevice.WebViewOne.sendData = {"type": "notification", "data":
                notification};
      }

      if(currentDevice.type === "adapter.lifx.bulb.light"){
            currentDevice.lightOne.setColor = {"hue": 319, "sat": 100, "brightness": 100,
                "kelvin":0};
      }

      if(currentDevice.type === "android.base.audio"){
            currentDevice.AudioOne.startAudio = "alarm.mp3";
      }
      ...
}
```

3. **Render Notification Data for Output Devices**
   The final step in our pipeline is about the rendering of notification data for the different output devices, we collected in step 2d). The whole iteration process can be seen in listing 4.3. For each device in our list, we have to check which module has been configured for the current device, so that we can call a specific function which can be used to send the notification data to the device. This check is necessary and it prevents, that functions are called which are not available for a module and that the output device always receives the data in a correct and usable format. For example, a LIFX bulb can't do anything with the notification object, it can't display the text or the picture of the sender. It can only display a color based on different parameters and therefore we have to create and send a new JSON object to it which describes this color. Table 4.2 gives an overview about the different device types and how the notifications are rendered for them.

4.4.3 Synchronization between Server and Android Client

Each notification that has been posted on an Android device is forwarded to the server and saved in the global notification list via the scripts. If the user dismissed a notification, the server receives it via a new event and deletes the notification from it's list. This should be ordinary procedure, but it's possible that the user dismissed a notification outside of his local meSchup network or that a UDP datagram got lost. Consequently, the server never received the event and therefore can't delete the notification from its

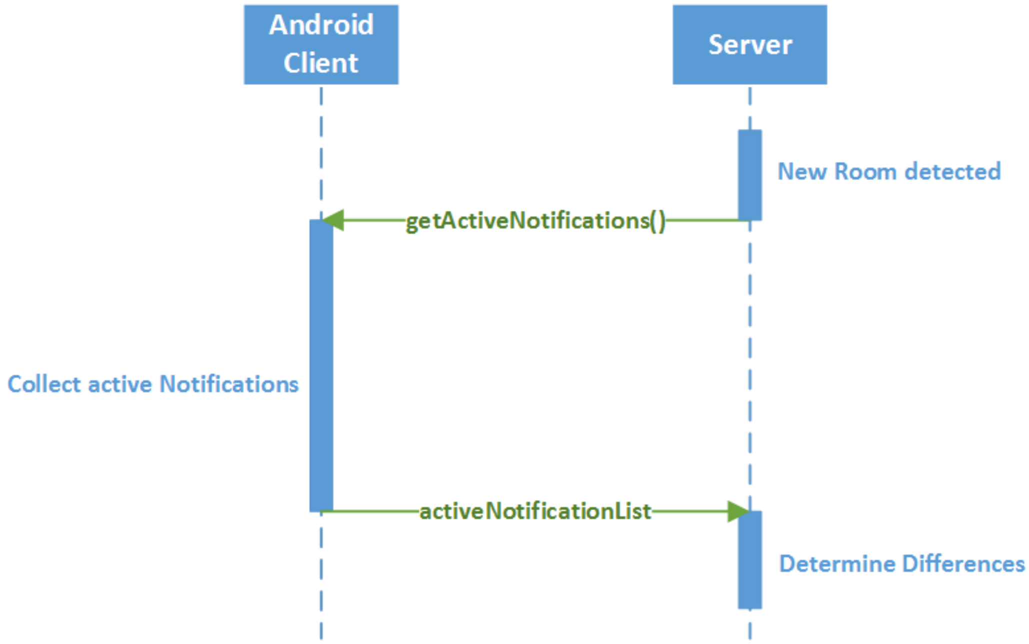


Figure 4.12: Sequence diagram for the synchronization process between the server and the Android client.

list. The notification would circulate inside the system and each time the user connects to the network and enters a new room, the notification would be displayed on every device in that room. Besides the restart of the whole system or the deletion of the whole notification list, which has the side effect of removing the notifications from other users too, there is no other possibility to remove this notification.

To prevent such situations, we implemented a synchronization task between the server platform and the Android app. Every time the user enters a new room, a request is sent to the user's phone, which triggers a function that collects the data of all active notifications on the phone and sends them back to the server. The server can now compare its local notification list with the list from the user and determine any differences. Figure 4.12 illustrates this process in an UML sequence diagram.

We can distinguish two cases here: if the notification list on the server side is longer or has the same length as the client side list, then it is very likely that at some point the server did not receive an dismissed Event triggered of a notification and therefore the server list contains notifications that are outdated. In this case, we update the server list and delete the outdated notifications. In the second case, the client list is longer and contains notifications that never received the server, because they were received outside of the meSchup network, so the server was never informed about them. We treat these notifications as if they had just been posted by an app and add them to our server list, so that they can be displayed on an output device.

5 User Study

This chapter will present the details about our performed user study. We will first explain the evaluation process before we come to our results and discussion.

5.1 Evaluating the System

In order to evaluate the notification system we conducted a two-week in-the wild study in which all participants were equipped with the same set of devices and installed the system for each participant at home. We wanted to know in which situations do users prefer which output devices or methods for which notification types. The following sections will provide details about the participants, the procedure, the used hardware setup, the tasks and collected data.

Participants

We collected 7 users, 2 female and 5 male, aged 20-61 (M=40, Mdn=27, SD=19.45) all with an academic education background. 2 of them study software engineering, 1 Slavic languages and 1 music. 1 person works as a doctor for internal medicine, 1 person works as a teacher and 1 person is deputy director of a engineering research institute. Regarding their technical background, everyone of them owns a smart phone, 3 persons a tablet, 5 persons a laptop, 4 persons a radio, 2 persons a desktop PC, a AV receiver or a stereo system.

When asked how many applications on their smart phones notify them, P1 guessed a number of *10*, P2 *7*, P3 *15*, P4 *3*, P5 *11*, P6 *2* and P7 *12* (M=8.57, Mdn=9.28, SD=4.79) and when asked about their general opinion about notifications, we observed an interesting pattern: the three participants that are over 60 years old answered uniformly, that notifications are not important for them and they don't really need them. These were also the participants that answered, that they use their smart phone less than 1 hour per day, feel more interrupted by notifications and guessed, that they would only receive less than 10 notifications per day. The "younger" generation uses the phone

User	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue
1	Yellow	Yellow	Yellow	Yellow	Green	Green	Green	Blue	Blue	Blue	Blue	Red	Red	Red
2	Green	Green	Green	Green	Blue	Blue	Blue	Red	Red	Red	Red	Yellow	Yellow	Yellow
3	Blue	Blue	Blue	Blue	Red	Red	Red	Yellow	Yellow	Yellow	Yellow	Green	Green	Green
4	Red	Red	Red	Red	Yellow	Yellow	Yellow	Green	Green	Green	Green	Blue	Blue	Blue
5	Yellow	Yellow	Yellow	Yellow	Green	Green	Green	Blue	Blue	Blue	Blue	Red	Red	Red
6	Green	Green	Green	Green	Blue	Blue	Blue	Red	Red	Red	Red	Yellow	Yellow	Yellow
7	Blue	Blue	Blue	Blue	Red	Red	Red	Yellow	Yellow	Yellow	Yellow	Green	Green	Green

Table 5.1: Schedule of the user study. *Yellow* = LIFX bulb, *Red* = voice output, *Blue* = Visual list design, *Green* = Visual bubble design.

more often per day (1 person less than 1 hour, 1 person 1-2 hours, 2 persons 2-5 hours), receives 20-30 (3 persons) or 30-40 (1 person) notifications per day and sees notification as as a “helpful” and important mechanism to be informed about news or to stay in touch with other people.

Procedure

Over a period of two weeks, the participants were asked to use the system, what was rewarded with 20 EUR cash. In order to be able to participate in the study, everyone needed only a smart phone with at least Android 4.3 because all other devices were provided by the university. Table 5.1 shows the used schedule. Every participant received a unique number, that allowed us to identify his personal schedule regarding the output designs. Via our scripts on the server side, we could change to a new output method after three or four days, so that every participants had the possibility to experience them on at least one day of the weekend. Each participant received a comprehensive briefing regarding the different devices, the Android app and the possible configurations for their favorite apps and the forwarded data fields. The participants were also instructed, not to switch off the devices in order to prevent possible problems with the displaying of their notifications. However, only the lamp could be turned off via the official app, so that it still could be used to display a new notification.

The tablet was configured to display the visualizations and to perform the voice output. For the visual part, we used the *list design* from Design D in Figure 3.3 and the *bubble design* from Design G in Figure 3.4. For the lamp and the voice output we created a wide list of configurations in our server sided scripts, that covered some of the most used applications. For each new notification, the bulb first blinked in our configured color and then changed back to its previous color (see appendix A.4 for the complete list of all

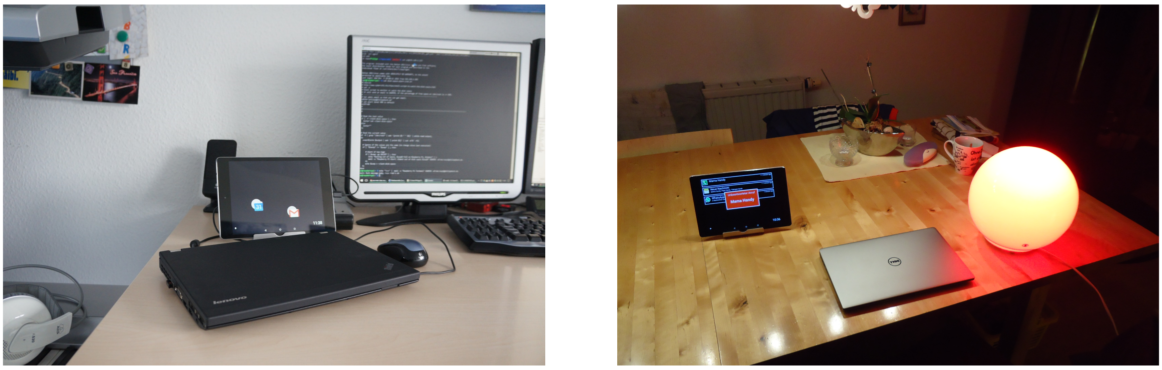


Figure 5.1: Two examples, how the participants positioned the given hardware setup.

app names and the used LIFX color). This process happened two times and also worked, when the lamp was turned off via the official LIFX application. For the voice output we only used the notification *title*, *text* and *textLines* fields to create a custom string, like *"New message from "+notification.title+":"+notification.text*. When no title or text fields are forwarded the string changed to *"New message from "+notification.appName*.

Hardware Setup

Each participant received the same set of hardware devices: one Raspberry PI was used to host the meSchup platform and a second one was configured with a BLE scanner module to detect Bluetooth signals. On the output side, we used a white, round glass lamp with a diameter of 25cm, that served as a socket for a LIFX bulb and a Nexus 9 tablet from HTC running Android 7.0. The UP2 fitness tracker from Jawbone was used as a BLE sender and to identify, if a user is in the scanner room or not. The setup and our Android app was installed and configured for every participant at their home. Everyone could freely decide about the positioning of the devices and configure the color of the lamp via the official LIFX application. Figure 5.1 shows two examples in an office and dining room.

Tasks

For the study the participants had to use and monitor the installed notification system in their home environment and complete the given questionnaires after a new output method had been deployed. Certain abnormalities or improvement ideas could be recorded in a diary.

Data Collection

The following quantitative data was collected from questionnaires and log files that were recorded through the Android app.

- **Demographics:** We asked the participants about their age, gender, highest level of education and their current profession.
- **Mobile Phone Usage:** The participants answered questions regarding their daily smart phone usage and their general opinion on notifications.
- **Rating of the Output Methods:** After a new output method had been deployed automatically, the participants were asked to fill out a questionnaire in which they had to rate the former output method regarding the provided information content and intrusiveness factor and what they liked or disliked about it.
- **User Interactions with the Android App:** Every time a user selected a given check box, pressed a button, changed his color or used a switch in the Android application, we created a new entry in our log file.
- **Notification Data:** Every notification object that was posted during the study was recorded in our log file too. The log entry included different fields, like the post time, the name of the app that posted the notification, if a image was included or if it was a grouped notification or not. The title and text fields were anonymized and we only saved the length of each string.

Qualitative data was gathered via the questionnaires about each output method. For each method we asked which notifications the participants wanted to display on it and why. They could also suggest improvement ideas and describe the situations in which they received notifications, how many people were present and if the current output method was appropriate for this situations or not. In a personal interview after the study we asked about a general rating of all output method, which notifications were the most important to them and which output method they would choose for them. We showed three example notifications and asked if the participants would like to display them in their environment, on which output method and if they would also display them, if another person is in the same room. Participants could provide ideas how the different methods could be combined, in which rooms in their home they want to be informed about notifications, during which activities they do or do not want to be notified and if they would like to use the system again.

5.2 Results

By the end of the study, we asked the participants to send in their log files that recorded the necessary information about the posted notifications and analyzed them together with the filled in questionnaires and the statements from the interviews.

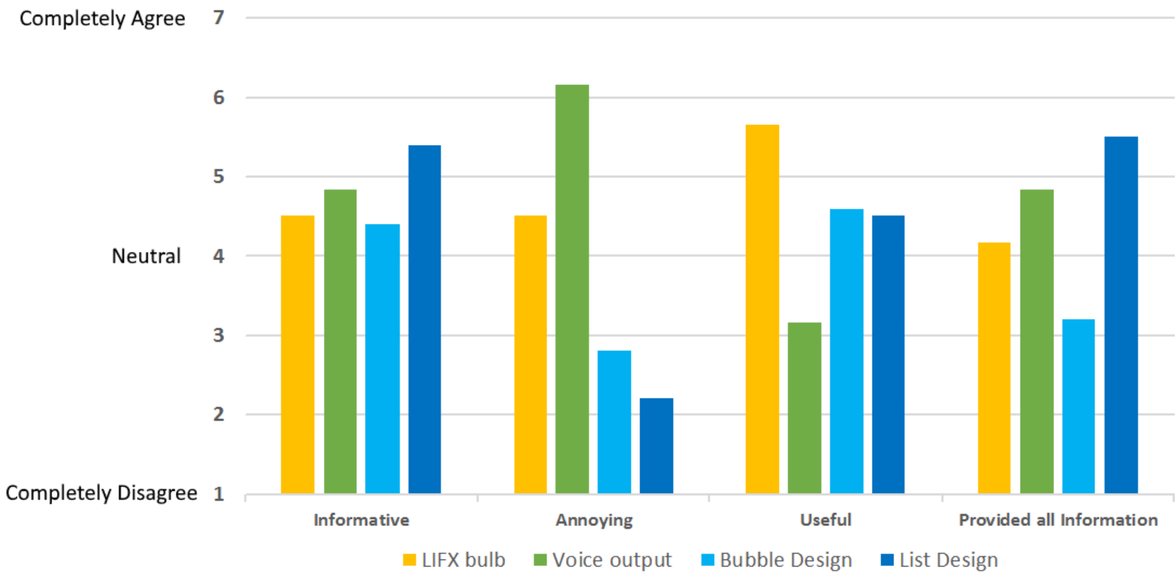


Figure 5.2: Questionnaire results to the statements, *"The used output method was very <informative|annoying|useful>."* and *"The used output method provided all necessary information."* on a Likert scale from 1 (*Completely Disagree*) to 7 (*Completely Agree*). The results are grouped for each output method.

5.2.1 Quantitative Results

Figure 5.2 shows the ratings for the four output methods regarding how informative, annoying or useful they were and if they provided all the necessary information about an incoming notification. The list design was rated as the output method that is the most informative (M=5.42, SD=1.71) followed by voice output (M=4.83, SD=0.75), LIFX bulb (M=4.5, SD=1.51) and the bubble design (M=4.4, SD=1.81). The participants rated the voice output with the highest annoyance factor (M=6.16, SD=0.98) followed by the LIFX bulb (M=4.5, SD=1.64), the list (M=2.28, SD=0.48) and bubble design (M=2.8, SD=1.48). Compared to the other output methods, the LIFX bulb was rated as the most useful (M=5.6, SD=0.51) followed by the bubble (M=4.6, SD=1.51) and the list design (M=4.5, SD=1.61). The voice output was rated as the least usefulness method (M=3.16, SD=1.32). When asked if the used output method provided all the

App name	Category	Percentage
WhatsApp	SMS/IM	36.41
Google App	system	5.87
Messenger	SMS/IM	4.19
E-Mail	email	3.61
Gmail	email	2.64

Table 5.2: Top 5 apps that generated the most notifications per day excluding apps with a flooding behavior.

necessary information about a notification, participants rated the list design (*M*=5.57, *SD*=1.27) as the highest, followed by voice output (*M*=4.83, *SD*=1.32), LIFX bulb (*M*=4.16, *SD*=1.47) and bubble design (*M*=3.2, *SD*=1.64).

Log file data

During the study, which lasted from 1st to the 15th of November 2016, we collected *22,452* notifications from our seven participants. However, this number is not very meaningful, because it also includes notifications from apps that "flood" the system with a large number of notifications in a short period of time, for example when a download happened in the background or when the system informs about a low battery. Table 5.2 shows the five apps from the overall list that posted the most notifications per day excluding apps with the flooding behavior.

From this total number, only *6348* (28.2%) notifications were allowed to be forwarded. If we divide this number by the number of participants and days, we gain an average of *64.77* notifications per user per day during our study. We also wanted to know, to which categories the forwarded and non-forwarded notifications belong and categorized the corresponding apps into *tool*, *social*, *news*, *game*, *email*, *calendar/reminder*, *shopping*, *media* and *SMS/IM* [37].

For *21* apps (see appendix A.2 for the full list and the assigned categories) the participants allowed the forwarding of notifications. We selected the five categories with the most app entries and did the same for the *63* apps that were not allowed for forwarding (appendix A.3). Because our checkboxes for the forwarding behavior can bee seen as some kind of "white-list", all apps that were not actively configured for forwarding are put into the non-forwarding category. Figure 5.3 shows the top five categories for each data set. For the non-forwarded apps we had 3 app entries belonging to the category media, 0 to email, 7 to SMS/IM, 4 to news, 18 to tool and 18 to system. When we look

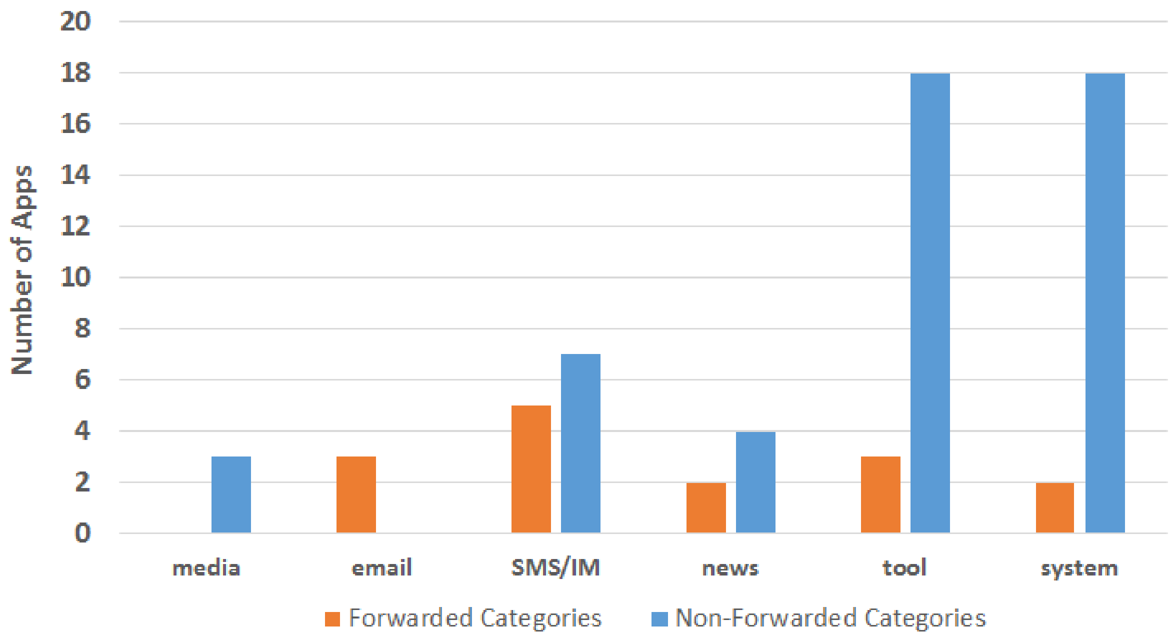


Figure 5.3: Top 5 most forwarded and non-forwarded app categories.

at the forwarded app entries, we can see 0 entries in the category media, 3 in email, 5 in SMS/IM, 2 in news, 3 in tool and 2 in system.

Apps in the category email were always forwarded while apps in the category media were never forwarded. *7* app entries in the category SMS/IM were the most forwarded and *18* entries in the categories tool and system were the most non-forwarded. Some apps that belong to this categories show the mentioned flooding behavior and could therefore be ignored by users, meaning that the forwarding of notifications was not allowed. It is important to mention here, that from the 36 tool and system entries, only 7 apps generated over *80%* of the notifications in these two categories.

5.2.2 Qualitative Results from Questionnaires

When asked about the most important kind of notifications the participants wanted to display, all of them mentioned *incoming calls*, *e-mails* and *text-messages "to stay in touch with friends and family members"* (P5). Notifications were mainly displayed in the evening in situations, when a person was either laying on the couch and sleeping, watching TV or reading a book, during the supper while talking with the family or working at the desk. For the list and bubble designs as well as the lamp, all participants answered unanimously that the presentation of the notifications on these output methods was appropriate in these situations, because they were *"unobtrusive"* (P2), *"It's subtle,* *but visible enough from the corner of the eye."* (P1) or that they could just ignore the presentations if they didn't want to be interrupted (P2, P3, P5). The participants disliked the voice output in the above situations, because they felt *"interrupted"* during another task (P5), while practicing the piano (P4) or woke up while sleeping (P6).

When asked how many people were present when the notifications were received, 2 participants said they were alone, 3 participants had one person around them, 1 participant had 2 persons around him and 1 participant was surrounded by 3 persons. 2 Participants now rated the lamp as inappropriate, P2: *"When friends were over, it was distracting."*, while the opinions about the visual output designs remained unchanged (all participants) and the voice output was rated as *"distracting"* (P2) and *"annoying"* (P1) when having a conversation.

LIFX bulb

Participants liked the LIFX bulb overall, because it was used as a *"silent"* (P1, P3, P6) method to inform about notifications and that it could also be used in situations, were one didn't hear the ring tone of the smart phone, *"Sometimes I can't hear the Smartphone and so the bulb informed me"* (P7). Some disliked, that it didn't provide much information about a notification (P2, P5): *"For some notification I'd like to see hints of the content, which is not possible with the light"* (P2) or that they perceived the light *"as uncomfortable during the night"* (P6) and *"Sometimes the blinking was too interruptive"* (P1) or "the light too bright" (P3). One participant suggested a possible improvement: users could assign specific lamp colors to certain people via the Android app, so that it is possible to distinguish the notifications from these people based on the displayed color. Another suggestion included the idea, that the displayed color that has been assigned to a certain app should not displayed instantly, but rather fade in over time to provide a smoother experience.

Bubble Design

The opinions about the bubble design were very divergent and different: P2 wrote *"The output method is not very intrusive. Meaning that I can just ignore it and it is not very annoying."* and that he could see with a glimpse *"how much was going on by the size of the bubble"*, but he disliked *"that for single messages I did not see a hint of the content."*. P1 had a very different opinion about the design, because there was *"little information and inefficient use of screen space"* and the *"fast-moving bubbles are annoying"*. P5 suggested, that the design could be improved by displaying more information, for example who was the sender of a notification or an excerpt of the text.

List Design

The rating results about the list design are also reflected in the received comments. Participants liked it because it gave them all the necessary information about a notification (P1, P2, P7) and that it was possible to see from distance the sender and the written text (P5) combined with a low intrusive factor (P3, P4). P1 wrote: *"It's subtle, but visible enough from the corner of the eye."* and for P7 *"there is no better Information compared with the native smart phone."* P1 also mentioned, that *"it serves as a replacement for popups"* but he disliked that *"it's an extra device that consumes power and uses desk space."* Overall, there were no major suggestions for improvement. Only P6 would have wanted a larger font and P1 suggested, that *"the screen should turn off when no users are present to save power."*

Voice Output

The voice output was rated as the most annoying output method. Participants disliked the loud voice (P1, P2, P3, P6), while two participants also felt frightened about it (P3, P6) or interrupted (P1, P5). P4 liked, that *"you are directly informed about the sender and can already estimate the importance of the message"* and for P1 it *"Gives enough information, so that checking the actual message/notification becomees unnecessary"* but P1 also mentioned, that *"speech output should absolutely be turned off during phone calls."* P5 wrote, that she was well informed about everything, but felt interrupted during ongoing tasks. Participants also provided some improvement ideas: P1 and P5 would like to shorten the read text, so that the computer voice would be less annoying. P1 thought of a feature, similar to the various digital assistants that can be found in the systems from Amazon, Google, Apple and Microsoft. If the user says *"what's going on?"*, the system could *"read out the last notification or only read out the sender's name or app name and then deliver more information when the user asks to hear more."*

5.2.3 Qualitative Results from the final Interviews

During an interview (see appendix A.7 for the asked questions) after the study we asked every person, how a general rating of all four output methods would look like for her or him. All participants rated the visual output methods higher than the voice output (which is on the last place for six from seven persons) and for all of them people are text and email messages the most important notifications they want to display in their environment. All persons answered that they want to see the sender's name from an email, 3 the title and 2 an additional text line. 6 persons want to see the sender's name for text messages, 1 person wanted see an image of the sender instead and 2 people also wanted to see the message text. The participants had different ideas regarding the output methods: P1 and P4 would like to use the lamp and the voice output, P3 only the lamp, P6 only the list design and P2, P5 and P7 would like to see their notifications on the lamp supported by the list design.

When asked if they want to display the notification examples from Figure A.1, they answered uniformly that example A is an important reminder that shouldn't be missed, example B an unimportant information and example C should be displayed for 6 participants, but only for 3 of them with the contained image. When we showed the same examples and asked, if they should be displayed in situations were other people are present, 2 participants changed their opinion for example A and didn't want to display it: P2: *"I don't want other people to see my private appointments."* Example B and C remained unchanged for all of them, but P6 didn't want to display the image now, because *"private pictures are just for me and no one else."*

We also wanted to know, where they'd like to be informed about new notifications the most: the kitchen, living and dining room are all important to them whereas they want to be undisturbed in the bed and bath room. All participants can imagine, to combine the lamp with the list design in order to be informed in a discreet way and then to re-read any details if they are required. In our final question we asked, if they would like to continue using the system and why. 4 people negated this question since they see no need in it or there are not enough important notifications to be displayed, *"no notification is that much important to me that I want to be informed immediately"* (P2) or *"I don't receive many notifications and my wife has something against all technical and blinking devices"* (P7). 1 participant would only use the lamp because *"I have enough displays in my room!"* (P1). The 2 persons who would continue using the system said: *"It is comfortable, if one does not always have to look at his phone"* (P5) and P6 said: *"The system makes me aware of notifications that I would otherwise not even notice."*

5.3 Discussion

Our average number of 64.77 forwarded notifications per user per day is nearly identical with the number Pielot et al. [26] found out (63.5). It would be interesting, how many of these notifications were actually presented on an output device, but our log data whether a person was in the scanner room or not are questionable and hardly available. For some participants we have only a small amount of data, meaning that the BLE scanning functioned probably in the beginning of the study, but after the three or four days there are no more entries in our files which could be caused by a faulty communication between server and android system. Another reason could be, that participants forgot to recharge the Jawbone after two days or left it in the same room although they were not in the room. For the system, the user was still in the room but in fact he wasn't.

An interesting aspect is, that the number of SMS and instant messaging apps is nearly equal in the forwarding (*5*) and non-forwarding (*7*) categories. This could mean, that some participants have multiple messaging apps on their smart phone, but only some of them, like *WhatsApp* and the *Facebook Messenger* are frequently used and that notifications that were posted by other messaging apps are not as much as important and therefore should not be displayed on an output device. Regarding the four output methods we couldn't detect any major differences. All participants set their favorite apps and their forwarding behavior at the beginning of the study and didn't change these settings. From this we can conclude, that if a participant wanted to be informed about a new WhatsApp message or an email, this person wanted this information always, independently if the current output method was the lamp or the bubble design.

Based on our findings, we can confirm the statements *"Notifications are for messaging"* and *"Important notifications are about people and events"* from Sahami Shirazi et al. [29], because our participants mainly wanted to be informed about text messages and emails. The results of the final interviews showed, that the picture we gained through the filled in questionnaires after a new output method has been deployed remained unchanged, meaning that all participants rated the visual output methods higher than the voice output (which is on the last place for six from seven persons) and this finding does not change when other people are present.

Due to the limited number of participants, as well as the associated costs for the hardware, our results may not be sufficiently detailed and expressive. All in all, we gained a good picture about the four output modalities and collected various improvement ideas that could be used and implemented in the next version of the system or as collected notes for other researchers and developers.

6 Summary and Future Work

This chapter will summarize the work of this thesis and present ideas for future work. Through the questionnaires and final interviews, we could collect numerous suggestions and improvements for the system, which will be summarized in the last sections.

6.1 Summary

This thesis presented a notification system that uses different output devices to display smart phone notifications. We first discussed the related work regarding notifications and their interruptive effect in working situations. User studies [4, 8, 17] have shown, that users have difficulties to resume a task after they have been interrupted by a notification. Various ambient information systems [2, 5, 24, 34] have been created over the last years, trying to integrate an additional output source for different information in the home or office environments of the users. We closed this section with a look at other notification systems and implementations how users could be located in an indoor environment using BLE devices.

For our concept of a notification system, we presented different ideas which output devices could be used in an home environment and how notifications could be displayed on them, respecting their capabilities and restrictions. Based on this results, we described our implementation that uses the meSchup platform [19] to connect various devices and control their behavior with simple JavaScript files.

To evaluate the system, we conducted an in-the wild user study over two weeks. Participants mostly wanted to display email and messaging notifications on the lamp or the visual designs, while the voice output provided a similar information content, but was rated as very annoying and disturbing. This observed behavior also didn't changed when one or more persons were present, showing that participants want to be informed about new notifications but in a discreet way. Due to the limited number of participants, as well as the associated costs for the hardware, our results may not be sufficiently detailed and expressive. All in all, we gained a good picture about the four output modalities and collected various improvement ideas that could be used and implemented in the next version of the system or as collected notes for other researchers and developers.

6.2 Future Work

A follow-up study with more participants could evaluate the system again but now with the focus on the multi-user aspect that was implemented in a very basic form in our system but wasn't used in our study. At least two persons could use the system in an apartment with multiple rooms and multiple output devices. The participants report how good they were able to identify and distinguish their own notifications from other notifications on a visual design or lamp and how disruptive it was when several people in the same room received notifications and displayed them on a device.

So far, the context of a user was only considered if he changed the setting for his ringer mode at his phone. Future work could focus on the collection of more data that describes the current context, so that the system automatically decides which output device it should use if multiple devices are available instead of broadcasting the notification to all devices in a room. For example if the user is on the phone, the voice output shouldn't be used to prevent confusion and instead use a lamp or display, that is in the same room. Another possible idea is, that the system postpones the presentation of the notification by a specified time or automatically presents it to the user after he finished his phone call.

6.2.1 More Options to customize the System

Almost all participants expressed the wish to customize certain settings by themselves. Previous work confirms that users want customization options [38]. Some participants wanted to assign a certain color of the lamp to an app, while others wanted to assign a color to a specific person, so that all notifications are visually presented in that color by the lamp. Other statements included the idea, to determine time frames when the system presents notifications. P6 was informed about a new email via the voice output while she was sleeping, while P4 was awakened due to the bright light of the lamp and both would have wanted such an option to disable the system for example during the night. This idea could be extended even further, so that it is not only possible to set these time frames, but also the output device in certain situations. The user can decide, which notifications shall be displayed on which device in which room. Thus, it would be possible to display notifications in the evening in the living room on the lamp while one is watching TV and in the morning the system would use the voice output in the restroom. P2 suggested, that devices with a vibration motor could use it as an additional output channel and the user could define vibration patterns that identify the posting app or the sender.

6.2.2 Interaction with the Output Devices

Our notification system could be extended in such a way, that the output devices are not only used to display notifications, but also as devices that can be used to interact with the system itself. P1 thought about the idea, that the system could react to voice commands and for example reread the last received notification or all notifications from a certain person. These voice commands could also be combined with the Android app, so that the system creates a new notification on the user's phone as some reminder. With this extensions, the system would gain similar abilities like the various digital assistants that can be found in the systems from Amazon, Google, Apple and Microsoft. *Amazon Echo*, *Google Now*, *Siri* and *Cortana* can be used to start web searches, inform about the local weather or the stock market and remind the user about upcoming calendar events. Display devices with a touch screen could be used to dismiss notifications directly or to refresh and change the current visualization. Therefore, the interaction with notifications would work in both directions, from the phone to the display and vice versa. Of course, this would also lead to new problems, for example how could one prevent, that a user does not dismiss the notifications of another user?

6.2.3 Reminder Functions

The devices could also be used as a reminder, for example if a user has missed an email or text message, the lamp could blink in a specified pattern or color. In a scenario, in which the user pushed a pizza in the oven, went to another room and didn't take his smart phone with him, a reminder on the lamp could be useful, given that he started a timer on his phone beforehand. P6 thought about a reminder function for appointments. The user specifies how many minutes before an appointment or calendar event he wants to be reminded and on which output device. The lamp could then blink in the chosen color or the voice output reads a given text.

A Appendix

A.1 Notification Field Names

Description texts from Google API for StatusBarNotification[1] and NotificationCompat[2]

Field name	Data type	Description
appName	String	The name of the app that posted this notification.
packageName	String	The package name of the app that posted this notification.
postTime	long	The time in milliseconds when the notification was posted.
category	String	Categories can be used to describe the content of a notification. Examples for this field: ALARM, CALL, EMAIL, ERROR, EVENT, MESSAGE, SOCIAL, SYSTEM.
priority	int	The priority of a notification can be used as an indicator, how important the content of this notification could be for the user.
key	String	A unique key for notification that can be used as an identifier.
isClearable	boolean	Indicates if a notification can be cleared by the user or not.
isOngoing	boolean	Indicates the ongoing status of a notification.
number	int	Integer that indicates the number of events represented by this notification.
nid	int	A simple identifier for this notification.
notificationStatus	String	A notification can either be "new" or "removed".

[1] Google Android API StatusBarNotification https://developer.android.com/reference/android/service/notification/StatusBarNotification.html

[2] Google Android API NotificationCompat https://developer.android.com/reference/android/support/v4/app/NotificationCompat.html

ledARGB	int	Color of the phone LED.
ledOff	int	The number of ms for the LED to be off while it's flashing.
ledOn	int	The number of ms for the LED to be on while it's flashing.
color	int	Accent color to be applied by the standard Style templates when presenting this notification.
people	String	A String containing the people that this notification relates to, each of which was supplied to.
group	String	Get the key used to group this notification into a cluster or stack with other notifications on devices which support such rendering.
isGroupSummary	boolean	Get whether this notification to be the group summary for a group of notifications.
actionCount	int	Get the number of actions in this notification in a backwards compatible manner.
isLocalOnly	String	Get whether or not this notification is only relevant to the current device.
visibility	int	Sphere of visibility of this notification, which affects how and when the SystemUI reveals the notification's presence and contents in untrusted situations (namely, on the secure lockscreen).
appIconPath	String	URL for the appIcon image.
senderIconPath	String	URL for the senderIcon image.
imagePath	String	URL for all other images that are nested inside the notification.
title	String	The title of the notification.
titleBig	String	Title of the notification when shown in expanded form.
tickerText	String	Text that summarizes this notification for accessibility services.
text	String	The main text line.
textBig	String	This is the longer text shown in the big form of a notification.
textInfo	String	Small piece of additional text.
textSub	String	Third line of text.

textSummary	String	This is a line of summary information intended to be shown alongside expanded notifications.
textLines	String	An array of multiple text lines. Each line ends with a new line character.
ringerMode	String	Contains the current setting of the ringer mode, possible settings: normal, vibrate, silent.
doNotDisturbMode	String	Contains the current setting of the do not disturb mode, possible settings: disabled, important, silence, alarms.

A.2 Most forwarded Apps

App name	**Category**	**Percentage**
WhatsApp	SMS/IM	56.21
Telefon	system	13.11
Google App	system	6.76
E-Mail	email	5.58
Messenger	SMS/IM	4.69
Gmail	email	4.08
Clock	system	2.25
Uhr	system	2.21
Mail	email	1.61
LIFX	tool	0.80
VVS Mobil	tool	0.76
Nachrichten	news	0.63
IRCCloud	SMS/IM	0.33
SMS/MMS	SMS/IM	0.27
X10Manager	tool	0.24
ZEIT ONLINE	news	0.22
N26	tool	0.11
Skype	SMS/IM	0.09
Kalender	system	0.03
Anrufverwaltung	system	0.02
Instagram	social	0.02

A.3 Most non-forwarded Apps

App name	Category	Percentage
Amazon Kindle	media	24.58
Amazon Music	media	18.94
Android system	system	21.35
AntiVirus	tool	7.06
App-Shop	system	6.64
Assist	system	4.93
Avast Mobile Security	tool	2.88
Chrome	system	2.61
Dialler	tool	1.40
Download Manager	system	1.04
Download-Manager	system	0.92
Dropbox	tool	0.91
Einstellungen	system	0.86
File Manager	system	0.70
Flipboard	SMS/IM	0.66
Fotos	system	0.56
Galaxy Apps	system	0.51
Google App	system	0.34
Google Play Musik	system	0.33
Google Play services	system	0.29
Google Play Store	system	0.27
Google Play-Dienste	system	0.22
Google Text-in-Sprache	system	0.18
Instagram	SMS/IM	0.17
IRCCloud	SMS/IM	0.16
Kalender	calendar/reminder	0.15
Kamera	system	0.14
LIFX	tool	0.14
Mail	email	0.12
Maps	system	0.10
Messaging	SMS/IM	0.09
Messenger	SMS/IM	0.09
moovel	tool	0.07
MortPlayer Music	media	0.06

Musik	media	0.06
Nachrichten	news	0.06
Oberfläche	tool	0.05
OMA Client Provisioning	tool	0.03
Opera	tool	0.03
PageBuddyNotiSvc	tool	0.02
phase6	tool	0.02
Pinterest	tool	0.02
Pokemon GO	game	0.02
Pushbullet	tool	0.02
Quizduell	game	0.02
S Planner	tool	0.02
Skype	SMS/IM	0.01
Small Apps-Manager	tool	0.01
Speicher	system	0.01
SPIEGEL ONLINE	news	0.01
System UI	system	0.01
System-UI	system	0.01
tagesschau	news	0.01
Taschenlampe	tool	0.01
Telefon	system	0.01
Uber	SMS/IM	0.01
UC Browser	tool	0.01
Update Center	tool	0.01
VNC Viewer	tool	0.01
Wetter- & Uhr-Widget	tool	0.01
X10Manager	tool	0.01
YouTube	media	0.01
ZDFheute	news	0.00

A.4 Most used Apps and their Color Configuration

Name	**packageName**	**HSV color for LIFX**
Android Email	com.android.email	(60,100,100)
Android Phone	com.android.dialer	(176,100,100)
Dropbox	com.dropbox.android	(248,100,100)
Facebook	com.facebook.katana	(248,100,100)
Facebook Messenger	com.facebook.orca	(248,100,100)
Gmail	com.google.android.gm	(319,100,100)
Google+	com.google.android.apps.plus	(319,100,100)
Google Hangouts	com.google.android.talk	(120,100,100)
Google Play Music	com.google.android.music	(9,100,100)
Google Play Store	com.android.vending	(9,100,100)
Instagram	com.instagram.android	(9,100,100)
Skype	com.skype.raider	(248,100,100)
Snapchat	com.snapchat.android	(60,100,100)
Spotify	com.spotify.music	(120,100,100)
Twittter	com.twitter.android	(248,100,100)
VVSMobil	com.mdv.VVSMobil	(9,100,100)
WhatsApp	com.whatsapp	(120,100,100)
Default Color	-	(260,100,100)

A.5 Demographic Questionnaire

1. My participant number is...
2. How old are you?
3. I identify my gender as...
    a) Woman
    b) Man
    c) Not binary
    d) No answer
4. What is your highest level of education?
    a) No education

    b) Primary School
    c) Middle School
    d) High School
    e) Diploma degree
    f) Bachelor's degree
    g) Master's degree
    h) Doctorate degree
    i) Professional degree

5. What is your current job and in which field?
6. Which devices do you own?
    a) Smart Phone
    b) Tablet
    c) Smart TV
    d) Smart Watch
    e) Stereo System
    f) Desktop PC
    g) Laptop
    h) Radio
    i) Fitness Tracker
    j) AV Receiver
7. How many hours do you spend each day with your smart phone?
    a) less than 1 hour
    b) 1-2 hours
    c) 2-5 hours
    d) 5-10 hours
    e) more than 10 hours
8. What is your general opinion about notifications?

9. How many notifications do you receive per day?
    a) less than 10
    b) 10-20
    c) 20-30
    d) 30-40
    e) 40-50
    f) more than 50
10. Do you use applications that process notifications, e.g. Notification Mirroring?
11. How many apps on your smart phone notify you?
12. Do you feel interrupted by notifications? (Likert scale from 1 to 7)
13. How many hours do you usually spend at home on a working day?
14. How many hours do you usually spend at home on the weekend?

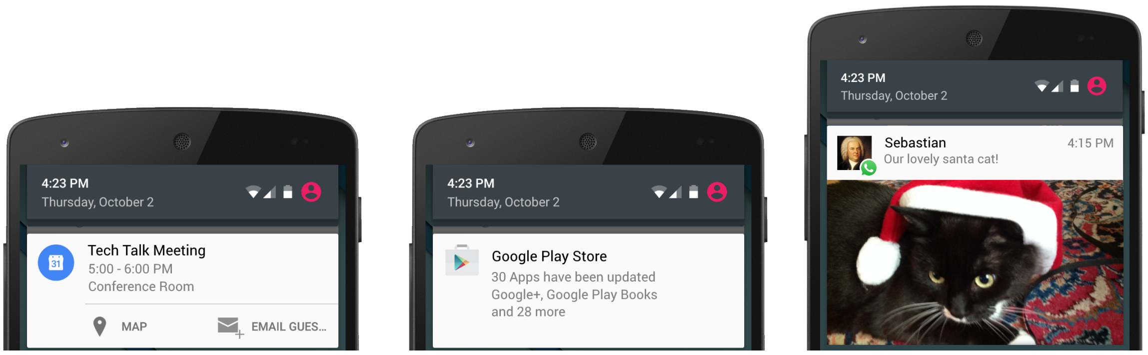
A.6 Questionnaire for the Output Methods

1. My participant number is...
2. I used the following output method the last days
    a) LIFX bulb
    b) Tablet - BubbleDesign
    c) Tablet - ListDesign
    d) Tablet - Speech Output
3. Please rate the following statements (Likert scale from 1 to 7)
    a) The used output method was very informative
    b) The used output method was very annoying
    c) The used output method was very useful
    d) The used output method provided all necessary information about a notification I needed

4. Please describe your most important kind of notifications and why you displayed them on the used output method
5. I liked/disliked the used output method for the following reasons
6. The output method could be improved for the following reasons
7. In which situations did you receive the notifications?
8. In these situations the presentation of the notifications was appropriate
9. In this situations I liked/disliked the following about the used output method
10. How man notifications did you perceive in this situations?
11. How many people were present when you received the notifications and who were these people (family members, postman etc.)?
12. Was the used output method appropriate while other people were present?

A.7 Interview questions

1. How would you rank the different output methods, which you liked the most, the second most (mention some advantages and disadvantages)?
2. What kind of notifications do you want to be displayed in your environment and what type of output method would you choose? What information would be especially important to you?
3. Notification examples (Figure A.1)



Example A Example B Example C

Figure A.1: Notification examples.

a) Would you like to display this notification in your environment? Why or why not?

b) Which output method would you choose? Why would you choose this method?

c) What information would be especially important to you? Why do you want to see this information?

4. Activities

   a) When were the notifications displayed in your environment?

   b) Were there any moments when you found the notifications annoying?

   c) Were there any moments when notifications were helpful to you?

   d) Were there any activities where you did not want to receive notifications? What were these activities? Why did you not want to be notified?

   e) Were there any activities you wanted to receive notifications? What were these activities? Why did you want to be notified?

5. Privacy

   a) How did you like the output method when other people were present?

   b) In which situations were other people present?

   c) For each mentioned situation: How shall notifications presented to you in this situation? (Examples from question 3)

6. Where would you like to see the notification presentations (Kitchen, living room, bedroom etc.)?

7. Can you imagine to combine the different output methods? If yes, why this combination? If no, why not?

8. What should a notification system provide for you at home, so you would use it?

9. Would you like to reuse the system? Why or why not?

10. Further improvement suggestions or comments

Bibliography

[1] F. Alt, A. S. Shirazi, T. Kubitza, and A. Schmidt. Interaction techniques for creating and exchanging content with public displays. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 1709–1718, New York, NY, USA, 2013. ACM.

[2] M. Altosaar, R. Vertegaal, C. Sohn, and D. Cheng. Auraorb: Using social awareness cues in the design of progressive notification appliances. In *Proceedings of the 18th Australia Conference on Computer-Human Interaction: Design: Activities, Artefacts and Environments*, OZCHI '06, pages 159–166, New York, NY, USA, 2006. ACM.

[3] R. M. Arlein, S. Betgé-Brezetz, and J. R. Ensor. Adaptive notification framework for converged environments. *Bell Labs Technical Journal*, 13(2):155–159, 2008.

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

[5] J. J. Cadiz, G. Venolia, G. Jancke, and A. Gupta. Designing and deploying an information awareness interface. In *Proceedings of the 2002 ACM Conference on Computer Supported Cooperative Work*, CSCW '02, pages 314–323, New York, NY, USA, 2002. ACM.

[6] Computerbase - Nicolas La Rocco. Android 7.0 Nougat im Test, August 2016. URL https://www.computerbase.de/2016-08/android-7.0-nougat-test/.

[7] F. Corno, L. D. Russis, and T. Montanaro. A context and user aware smart notification system. In *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*, pages 645–651, Dec 2015.

[8] M. Czerwinski, E. Horvitz, and S. Wilhite. A diary study of task switching and interruptions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '04, pages 175–182, New York, NY, USA, 2004. ACM.

[9] A. Dahley, C. Wisneski, and H. Ishii. Water lamp and pinwheels: Ambient projection of digital information into architectural space. In *CHI 98 Conference Summary on Human Factors in Computing Systems*, CHI '98, pages 269–270, New York, NY, USA, 1998. ACM.

[10] G. C. L. Delfa and V. Catania. Accurate indoor navigation using smartphone, bluetooth low energy and visual tags. In *Proc. 2nd Conf. on Mobile and Information Technologies in Medicine*, CHI '14, pages 1–4, 2014.

[11] Google. Android 7.0 for Developers, 2016. URL https://developer.android.com/about/versions/nougat/android-7.0.html.

[12] GSM Association. A Guide to Bluetooth Beacons, September 2014. URL http://www.gsma.com/digitalcommerce/wp-content/uploads/2013/10/A-guide-to-BLE-beacons-FINAL-18-Sept-14.pdf.

[13] H. Plischke and S. Schauer and A. Schülke and G. Kempter. Untersuchungen zu visuellen und psychophysiologischen Effekten von Fernsehen mit dynamischer Hintergrundbeleuchtung, May 2007. URL http://www2.philips.de/entspannter_fernsehen/downloads/GRP_Studienresuemee.pdf.

[14] R. Hansson, P. Ljungstrand, and J. Redström. Subtle and public notification cues for mobile devices. In *Proceedings of the 3rd International Conference on Ubiquitous Computing*, UbiComp '01, pages 240–246, London, UK, UK, 2001. Springer-Verlag.

[15] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen. The gator tech smart house: A programmable pervasive space. *Computer*, 38(3):50–60, Mar. 2005.

[16] S. T. Iqbal and B. P. Bailey. Oasis: A framework for linking notification delivery to the perceptual structure of goal-directed tasks. *ACM Trans. Comput.-Hum. Interact.*, 17(4):15:1–15:28, Dec. 2010.

[17] S. T. Iqbal and 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*, CSCW '10, pages 27–30, New York, NY, USA, 2010. ACM.

[18] B. Jones, H. Benko, E. Ofek, and A. Wilson. Illumiroom: Peripheral projected illusions for interactive experiences. In *CHI '13 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 869–878. ACM, April 2013.

[19] T. "Kubitza and A. Schmidt. *Towards a Toolkit for the Rapid Creation of Smart Environments*, pages 230–235. Springer International Publishing, Cham, 2015.

[20] T. Matthews, A. K. Dey, J. Mankoff, S. Carter, and T. Rattenbury. A toolkit for managing user attention in peripheral displays. In *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology*, UIST '04, pages 247–256, New York, NY, USA, 2004. ACM.

[21] A. Matviienko, M. Rauschenberger, V. Cobus, J. Timmermann, J. Fortmann, A. L"ocken, H. M"uller, C. Trappe, W. Heuten, and S. Boll. Towards new ambient light systems: a close look at existing encodings of ambient light systems. *IxD&A*, 26:10–24, 2015.

[22] D. S. McCrickard, R. Catrambone, and J. T. Stasko. Evaluating animation in the periphery as a mechanism for maintaining awarness. In *Proceedings of the IFIP TC.13 Conference on Human- Computer Interaction (INTERACT 2001)*, INTERACT 2001, pages 148–156, 2001.

[23] T. Miller and J. Stasko. Artistically conveying peripheral information with the infocanvas. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '02, pages 43–50, New York, NY, USA, 2002. ACM.

[24] H. Müller, M. Pielot, and R. de Oliveira. Towards ambient notifications. *Peripheral Interaction: Embedding HCI in Everyday Life*, 21, 2013.

[25] J. Park, J. Kim, and S. Kang. Ble-based accurate indoor location tracking for home and office, Nov 2015.

[26] M. Pielot, K. Church, and 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*, MobileHCI '14, pages 233–242, New York, NY, USA, 2014. ACM.

[27] C. Plaue and J. Stasko. Animation in a peripheral display: Distraction, appeal, and information conveyance in varying display configurations. In *Proceedings of Graphics Interface 2007*, GI '07, pages 135–142, New York, NY, USA, 2007. ACM.

[28] Z. Pousman and J. Stasko. A taxonomy of ambient information systems: Four patterns of design. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '06, pages 67–74, New York, NY, USA, 2006. ACM.

[29] A. Sahami Shirazi, N. Henze, T. Dingler, M. Pielot, D. Weber, and A. Schmidt. Large-scale assessment of mobile notifications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '14, pages 3055–3064, 2014.

[30] S.H.A. Begemann. A scientific study on the effects of Ambilight in flat-panel displays, July 2005. URL http://www.embedded.com/print/4012996.

[31] T. Skog, S. Ljungblad, and L. E. Holmquist. Between aesthetics and utility: Designing ambient information visualizations. In *Proceedings of the Ninth Annual IEEE Conference on Information Visualization*, INFOVIS'03, pages 233–240, Washington, DC, USA, 2003. IEEE Computer Society.

[32] M. Tomitsch, K. Kappel, A. Lehner, and T. Grechenig. Towards a taxonomy for ambient information systems. In *Workshop at Pervasive*, pages 42–47, 2007.

[33] TP Vision. Faszination Ambilight - Hintergrundinformationen, July 2016. URL http://www.philips.com/b-dam/corporate/newscenter/de/press-releases/personal_health/20160707_65PUS7601/Hintergrundinformationen_Faszination_Ambilight.pdf.

[34] M. van Dantzich, D. Robbins, E. Horvitz, and M. Czerwinski. Scope: Providing awareness of multiple notifications at a glance. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, AVI '02, pages 267–281, New York, NY, USA, 2002. ACM.

[35] M. H. Vastenburg, D. V. Keyson, and H. de Ridder. Considerate home notification systems: A user study of acceptability of notifications in a living-room laboratory. *Int. J. Hum.-Comput. Stud.*, 67(9):814–826, Sept. 2009.

[36] A. Voit, T. Machulla, D. Weber, V. Schwind, S. Schneegass, and N. Henze. Exploring notifications in smart home environments. In *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, MobileHCI '16, pages 942–947, New York, NY, USA, 2016. ACM.

[37] D. Weber. Notifications in a Multi-Device Environment. Diplomarbeit, Universität Stuttgart, Fakultät Informatik, Elektrotechnik und Informationstechnik, Germany, Januar 2015.

[38] D. Weber, S. Mayer, A. Voit, R. Ventura Fierro, and N. Henze. Design Guidelines for Notifications on Smart TVs. In *Proceedings of the ACM International Conference on Interactive Experiences for TV and Online Video*, TVX '16, pages 13–24, 2016.

[39] D. Weber, A. S. Shirazi, and 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*, MobileHCI '15, pages 1117–1122, New York, NY, USA, 2015. ACM.

[40] M. Weiser and J. S. Brown. Designing calm technology. *POWERGRID JOURNAL*, 1, 1996.

[41] C. Wisneski, H. Ishii, A. Dahley, M. G. Gorbet, S. Brave, B. Ullmer, and P. Yarin. Ambient displays: Turning architectural space into an interface between people and digital information. In *Proceedings of the First International Workshop on Cooperative Buildings, Integrating Information, Organization, and Architecture*, CoBuild '98, pages 22–32, London, UK, UK, 1998. Springer-Verlag.

All links were last followed on November 21, 2016.

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