Department of Electrical Engineering and Information Technology
Institute for Media Technology
Distributed Multimodal Information Processing Group
Prof. Dr.-Ing. Eckehard Steinbach

User Interfaces for Rule-Based Multimodal Interaction

Benutzeroberflächen für regelbasierte multimodale Interaktion

Maximilian Walker

Master Thesis

Author:	Maximilian Walker
Address:	[redacted]
Matriculation Number:	10707370
Professor:	Prof. Dr.-Ing. Eckehard Steinbach
Advisor:	Prof. Dr. rer. nat. Matthias Kranz, Dipl.-Medieninf. Andreas Möller
Begin:	02.12.2013
End:	02.06.2014

Department of Electrical Engineering and Information Technology
Institute for Media Technology
Distributed Multimodal Information Processing Group
Prof. Dr.-Ing. Eckehard Steinbach

Declaration

I declare under penalty of perjury that I wrote this Master Thesis entitled

User Interfaces for Rule-Based Multimodal Interaction
Benutzeroberflächen für regelbasierte multimodale Interaktion

by myself and that I used no other than the specified sources and tools.

Munich, June 1, 2014

Maximilian Walker

Maximilian Walker
[redacted]

Kurzfassung

Mobile Endgeräte ermöglichen heute die Einbeziehung zahlreicher Modalitäten in die Interaktion. Hinsichtlich der Ausgabemodalitäten ist visuelles, auditorisches und haptisches Feedback möglich. Auf der Eingabeseite wurden konventionelle Knöpfe durch Spracheingabe sowie Berührungs- oder Bewegungsgesten ergänzt. Weiterhin sind mobile Endgeräte mit einer stetig steigenden Anzahl von Sensoren gekennzeichnet, welche die Erfassung von Kontextinformationen ermöglichen (z.B. über Aufenthaltsort, Umgebungslicht und -lautstärke, Zeit oder Orientierung des Geräts).

Multimodalität und eine kontextsensitive Auswahl geeigneter Modalitäten können die mobile Interaktion durch das Ausgleichen von kognitiven oder Wahrnehmungseinschränkungen verbessern. Das Kombinieren mehrerer Modalitäten kann zu einer höheren Nutzungseffizienz und einem verbesserten Umgang mit Fehlern sowie mehr Flexibilität und Personalisierung führen.

Die aktuelle Situation für Nutzer mobiler Endgeräte erfordert einen manuellen Wechsel zwischen Modalitäten, was eine hohe Aufgabenlast nach sich zieht. Ein vollautomatisches System hingegen kann zu unerwünschten Modalitätsänderungen und ungenügender Transparenz für den Nutzer führen ("Black Box" Verhalten). Die Forschung im Bereich der multimodalen Interaktion konzentriert sich bisher mehr auf Modalitäten an sich als auf Konzepte, welche es dem Nutzer ermöglichen diese vorteilhaft einzusetzen.

Die vorliegende Masterarbeit untersucht einen regelbasierten Ansatz für Modalitätenwechsel, welcher einer geringeren Aufgabenlast für den Nutzer anstrebt ohne ihm dabei die Kontrolle über Modalitätenwechsel zu nehmen. Dieses Ziel soll mit Hilfe grafischer Benutzeroberflächen erreicht werden, die es dem Nutzer ermöglichen Regeln zu definieren, welche einerseits aus Kontextfaktoren oder Eingabeaktionen und andererseits aus Triggern zur Auslösung von Modalitätsänderungen oder Systemaktionen bestehen. Es existieren verschiedene Möglichkeiten, die es dem Nutzer ermöglichen solche Regeln in einer Benutzeroberfläche zu definieren und die Wahrnehmung von Modalitätsänderungen zu gewährleisten. Dazu wurden zahlreiche Benutzeroberflächen konzipiert, in einer Android Anwendung implementiert und abschließend hinsichtlich Effizienz, Effektivität und Zufriedenheit evaluiert. Eine der Benutzeroberflächen, welche durch ein situationsorientiertes mentales Modell des Nutzers gekennzeichnet ist, schnitt signifikant besser ab als konkurrierende Varianten. Die bevorzugte Art Wahrnehmung über Modalitätsänderungen zu gewährleisten stellte sich als sehr subjektiv heraus. Insgesamt wurde der regelbasierte Ansatz von den Studienteilnehmern sehr gut angenommen. Alle Teilnehmer gaben an, die entwickelte Anwendung gerne benutzt zu haben, über 95% empfanden sie als nützlich. Des Weiteren werden Erkenntnisse über das Verhalten von Nutzern und Meinungen hinsichtlich multimodaler Interaktion präsentiert, die in Interviews gewonnen werden konnten und statistische Daten vorgestellt, welche während der durchgeführten Feldstudie gesammelt wurden.

Abstract

Mobile devices nowadays allow the inclusion of multiple modalities when interacting with them. Visual, auditory and haptic feedback are possible output modalities. On the input side conventional hardware buttons have been complemented with speech input, as well as touch and motion gestures. Also mobile devices are equipped with an increasing amount of sensors which allow the collection of context information (e.g. about location, ambient light and noise level, time or device orientation).

Multimodality and a context-sensitive selection of suitable modalities can improve mobile interaction by overcoming perceptual and cognitive limitations users experience in varying situations. Combining multiple modalities can result in higher efficiency and improved error handling and allows for more natural interaction as well as more flexibility and personalization.

The current situation for smartphone owners requires manual modality switching which results in a high task load for the user. A fully automated system on the other hand can result in unwanted modality changes combined with insufficient transparency for the user ("black box" behavior). Research in the area of multimodal interaction has mainly focused on modalities per se, but less on concepts which allow the user to control these in a beneficial way.

The thesis investigates a rule-based approach to modality switching which aims to reduce the user's task load while leaving control about modality changes to the smartphone owner. This is accomplished through graphical user interfaces which allow the user to define rules containing context factors or user actions on the one hand and triggers evoking modality changes or system actions on the other hand. There are different possibilities to let the user create such rules in a user interface and to provide awareness about modality changes. To that end we conceived several user interfaces, implemented them in an Android application realizing rule-based interaction and evaluated them in a laboratory and a field study regarding efficiency, effectiveness and satisfaction. One user interface which is characterized through a situation-oriented user mental model performed significantly better than competing variants, while the preferred method to provide awareness about modality changes has shown to be subjective from person to person. All in all the rule-based approach was appreciated very well. All participants reported that they liked using the developed application, over 95% considered it useful. The thesis also presents insights about users' behavior and opinions on different aspects of multimodal interaction gained through user interviews and statistical data collected during a field study.

Contents

Contents vi

1. Introduction 1
1.1. Motivation . . . 3
1.2. Overview . . . 6

2. Background & Related Work 7
2.1. Mobile Interaction . . . 7
2.2. Modalities & Multimodality . . . 8
2.2.1. Output Modalities . . . 8
2.2.2. Input Modalities . . . 10
2.2.3. Multimodality . . . 11
2.3. Context . . . 12
2.3.1. Physical Environment . . . 12
2.3.2. Human Factors . . . 14
2.4. Related Work . . . 16

3. Concept 20
3.1. Rule-based Multimodal Interaction . . . 20
3.1.1. Goals . . . 21
3.1.2. Formalization . . . 22
3.2. Focus Group . . . 23
3.3. User Interfaces . . . 25
3.3.1. Representation and Definition of Situations . . . 25
3.3.2. Context Dialogs . . . 28
3.3.3. Output Modality Dialogs . . . 30
3.3.4. Icon Language . . . 31
3.3.5. Rule Layouts . . . 31
3.3.6. Navigation Flow . . . 32
3.3.7. Awareness . . . 32
3.3.8. Rule Creation Alternatives . . . 36
3.3.9. Input Rules . . . 36

4. Implementation 41
- 4.1. Framework 41
  - 4.1.1. Description 41
  - 4.1.2. Extending the Framework 43
- 4.2. Android Application 48
  - 4.2.1. User Interfaces 48
  - 4.2.2. Background Services & Evaluation Scheduling 54
  - 4.2.3. Laboratory and Field Study Versions 55

5. Evaluation 58
- 5.1. Laboratory Study 59
  - 5.1.1. Experiment Setup 59
  - 5.1.2. Participants & Demographic Data 59
  - 5.1.3. General Questions about Modality Usage 61
  - 5.1.4. Rule Creation 65
  - 5.1.5. Rule Creation Alternatives 72
  - 5.1.6. Interviews 73
  - 5.1.7. Awareness 75
  - 5.1.8. Rules for Input Modalities 78
  - 5.1.9. General Feedback 81
- 5.2. Field Study 83
  - 5.2.1. Experiment Setup 83
  - 5.2.2. Participants & Demographic Data 83
  - 5.2.3. Results 83
- 5.3. Discussion 87

6. Conclusion 90
- 6.1. Summary 90
- 6.2. Future Work 92

List of Figures 94

List of Tables 97

List of Acronyms 98

Bibliography 99

Chapter 1.

Introduction

We behave differently in varying situations: while it is appropriate to speak silently (or not at all) in a quiet environment such as a library, we speak with a loud voice when communicating in noisy public spaces like concerts. In some situations we even have to use other channels of communication such as gestures or facial expressions to make ourselves understood. Adapting our behavior to the current environment is so normal that we do it automatically without actively thinking about it. Not doing so would make communication impossible or in the other direction disturb other persons around us. The smartphone has nowadays become a constant companion in people's everyday life. Recent studies show that users keep their mobile phone at arm's reach 50% of the time and for even 90% of the time in the same room [1]. A typical day for many people begins by being woken up by the smartphone's alarm clock, afterwards common activities are reading the news on the smartphone during breakfast and taking photos with the built-in camera or looking for locations in a maps application during the day. On the way home in a car or public transport many people use their smartphone's music player. At night, many persons spend their time checking social networks or web sites on their smartphones before falling asleep. No matter where we go a mobile device is almost always with us.

In contrast to us, the smartphone does however not adapt its communication channels or *modalities* to the varying situations we encounter during a day. Almost everyone has experienced unpleasant situations when a smartphone starts ringing in a quiet environment or missed an important call in a loud surrounding. Unless we change its settings manually every time by using a variety of soft- and hardware controls it will behave in the same way all day. Thus, a device which was built to help us in everyday situations has to be taken care of. Varying situations can also require different forms of input when interacting with a mobile device. While it is easy to perform actions like unlocking the smartphone and looking for a particular app when being in a calm, steady position, this can be a difficult task when walking to the next bus station in a hurry.

But how can a smartphone know about the current situation of its owner and adapt its behavior accordingly? On the one hand modern smartphones are nowadays equipped with an increasing amount of sensors which can help to determine the current context. GPS receivers can determine a user's current location, light sensors and the built-in microphone are capable of recognizing the ambient light and noise level in the surrounding. With the help of accelerometers, gyroscopes and magnetometers it is possible to determine the mobile device's precise orientation.

On the other hand mobile devices nowadays allow the inclusion of multiple modalities when interacting with them. Apart from sound output, haptic feedback can be given through vibration, and the device's screen and LEDs enable visual communication with the user. On the input side conventional input methods like hardware buttons have been complemented with speech input, capacitive displays and orientation sensors are capable of recognizing touch and motion gestures. Using context information could help to improve the usage of these different in- and output modalities by selecting the right interaction channels for a given situation.

As foreseen by Tamminen et al. in 2004 [2] context-sensitive selection of modalities is nowadays "not [...] a distant dream" anymore through advancements in context management and recognition in mobile devices. Hakkani-Tür et al. [3] noted in 2011 that user-specific data is now "waiting to be exploited for "customized" user interaction instead of "one size fits all"" solutions. Imagine for example the following scenario:

> *Max wakes up in the morning. After breakfast he leaves his apartment and rides to university on his bicycle. His smartphone recognizes that he is currently biking and automatically enables vibration and sound feedback to make itself perceptible in his pocket even while moving through the loud morning traffic. When Max arrives at the library his mobile device automatically switches to visual notifications and disables sound output completely. In the afternoon Max is late for an appointment. Hurrying to the meeting point he takes his phone out of his pocket and quickly taps the display with both thumbs (like writing a message). His smartphone recognizes this gesture and immediately presents the messaging app. He writes a short message saying that he will be 10 minutes late. In the evening Max goes to a concert venue to listen to his favorite rock band. Even though the smartphone was still set to silent after the appointment, the device recognizes the high ambient noise level and automatically enables all output channels (sound, vibration and notification light). Thanks to the automatic adjustment Max realizes when a friend calls him to tell him that he is also at the concert. When he goes to sleep at night he places his smartphone on the bedside table with the display facing down. The smartphone recognizes this defined orientation and disables all sounds, vibration and visual notifications to avoid any disturbances during the night.*

1.1. Motivation

But how can the smartphone be told how to behave in particular situations? Relevant situations and usage preferences can be different from person to person so that it can be difficult to find a general solution for this problem area. This thesis investigates **graphical user interfaces** (GUIs) for a **rule-based** approach as one possibility to provide suitable modalities to the user. The idea behind this approach is that users can create rules containing context factors and/or user actions on the one hand and triggers evoking modality changes or other system actions on the other hand to describe complex contextual situations and control their mobile devices' behavior.

In the following multimodal interaction with mobile devices will be motivated from a user-centric point of view.

Perceptual Limitations

Chittaro [4] noted that physical parameters such as ambient light, noise, temperature and motion of a mobile user's context can be extremely variable, which can lead to perceptual limitations as one or more modalities are excluded: "For example, in a noisy street we can become unable to perceive sounds from the mobile device; under a glaring sun, we can be unable to discriminate color on the screen or even to read the screen at all, on a moving vehicle we might not notice vibrations generated by the device" [4]. A selection of suitable modalities based on identified usage situations by contrast allows to choose interaction channels according to the resources which are typically free in a situation [5]. Lemmelä et al. [5] collected and analyzed information regarding the characteristics, limitations, properties, and strengths of different input and output modalities. Based on this analysis they identified modalities best suited to overcome perceptual limitations in different situations (see Table 1.1).

Visual	Auditory	Haptic
Meetings, Office, Public places, Transportation	Driving, Sporting, Outdoor Situations (Sunshine, Gloves on, ...)	Public (noisy)

Table 1.1.: Modalities and modality combinations best suited for different situations [5].

Cognitive Limitations & Self-Management of Cognitive Load

Persons moving through changing environments have to be attentive to a permanent flow of events around them and respond to them with proper actions [4]. Their tasks, goals and social situations can quickly change based on the current context and especially mobile users are usually engaged in multiple activities simultaneously [5]. Even if we don't react to most events in our environment directly, we have to devote a particular amount of attention to them to decide whether actions are necessary or not. In contrast to calm environments (e.g. at home), mobile conditions can therefore make the interaction with a mobile device a secondary task which can not be given full concentration and cognitive resources [4]. A suitable selection of modalities could support the user in such stressful situations by offering interaction methods that appear comfortable in a given situation. Oviatt [6] observed that the possibility to choose between different modalities appeared to enable users to effectively self-manage their own cognitive load.

Impaired Motor Skills

Mobile conditions can lead to impaired motor skills for a smartphone user. Being located in a moving vehicle (e.g. a bus or metro) constantly de- and accelerating makes it hard to use a mobile device's keyboard or other input methods which require sensitive handling [4]. A mobile device can potentially compensate impaired motor skills by offering more robust interaction methods.

More Natural Interaction

Humans naturally interact with their environment through a complex composition of different interaction channels [7] or as Turk [8] puts it:

> "Human interaction with the world is inherently multimodal."

Employing multiple senses such as sight, hearing and touch (and sometimes also smell and taste) allows us to analyze a great variety of different properties of our environment [8]. Visual cues can be used to quickly identify other beings or objects or to capture characteristics of our environment through the perception of light and colors. Similarly audio cues can be used to estimate the distance to a source of noise or help to adapt our own voice and behavior to our environment. Tactile cues can be especially important if visual or auditory information is sparse (e.g. in darkness) and can be used to find targets and identify their nature (e.g. by recognizing surface structures or the temperature of an object). These environmental cues are naturally employed both sequentially and in parallel to prepare ourselves as good as possible to changing contexts [8]. Multimodal interaction with mobile devices bears the chance to bring a similar wealth of information to everyday smartphone usage. Dumas et al. [9] also emphasize the importance of this aspect by naming more natural ways of interaction and supporting and accommodating users' perceptual and communicative capabilities as the two main objectives of multimodal interfaces.

Efficiency & Improved Error Handling

When evaluating multimodal interfaces for an interactive maps application Oviatt [6] found that multimodal interfaces can speed up task completion by 10%. The author furthermore investigated the participants' error handling and could show that users made 36% fewer errors with a multimodal interface compared to a system with an unimodal interface. When asking the participants about their preferred way of interaction with the application over 94% stated to generally prefer multimodal interaction over unimodal interaction.

Flexibility and Personalization

Persons can be very different with regard to their preferred interaction techniques. While some people are, for example, very open to intelligent personal assistants and speech input, others might prefer more conventional ways to give input to their smartphones. Sound and vibration output is appreciated by many users to get instant feedback about new messages and or other events, while it can be distracting and too obtrusive for persons who spend a lot of time with concentrated working. Multimodality can allow for a more flexible use of input modes, including alternation and integrated use [8]. Offering more choices for in- and output leaves the user ample scope for flexible personalization [9].

Accessibility

Having the possibility to choose between different modalities does not only serve personalization purposes, but can also become a necessity for people with disabilities. These can include sensory and motor impairments, temporary illnesses, conditioned cognitive skills or language barriers. Multimodal interfaces can increase the accessibility of smartphones by permitting users to choose and control how they interact with a mobile device [10]. Oviatt and Cohen [10] mention, for example, that visually or manually impaired users may prefer speech input, while persons with a hearing impairment or a strong accent might prefer tactile input methods.

Another advantage of multimodalty with respect to physiological conditions can be seen in the possibility to alternate in- and output methods in extensive usage situations. Repetitive stress or physical damage (e.g. when using a keyboard for a lengthy period of time) can be avoid by alternating between or temporarily disabling different modalities [10]. Thus, multimodal interfaces may accommodate not only individual differences, but also permanent or temporary handicaps [8].

Social Circumstances

Chittarro [4] mentions social circumstances as a motivation for multimodality: in situations where using particular modalities would be well suited with regard to perceptual or motor limitations social norms applicable for different environments can make them unsuitable or unpleasant for others. Examples are conferences, where it is not tolerated to use auditory modalities, whereas using the smartphone silently is socially accepted or movie theaters where neither sound nor visual modalities are desirable, as a bright screen can disturb others spectators. Chittaro furthermore refers to the usage of gestures from a social point of view. Highly noticeable gestures may irritate other people in specific places such as police stations or hospitals and can appear provoking in environments where persons are oversensitive to aggressive cues (e.g. a group of hooligans) [4].

New Forms of Interaction

Oviatt and Cohen [10], as well as Dumas [9] see multimodal systems as a chance to facilitate new forms of interacting with computers. A recent example might be seen in the advent of capacitive touch screens or intelligent personal assistants (such as Apple's Siri [1], Google Now [2] or Microsoft's Cortana [3]). While formerly input was primarily constituted by hardware buttons, the inclusion of new tactile (touch screens) and auditory (speech input and recognition) modalities has revolutionized the way we interact with mobile devices.

1.2. Overview

The thesis is organized as follows. We begin with the theoretical background of mobile interaction, multimodality and context-awareness. Definitions and important aspects of the concepts *modality* and *context* will provide the basis for a formal description of rule-based multimodal interaction and the following chapters. After a literature research on existing multimodal toolkits and frameworks we describe our concept and the results of a focus group as the first step of the development process. The implementation, realized as an Android application, will then be explained in detail to conclude the development phase. The created user interfaces and the developed mobile application were evaluated in a laboratory and a field user study. All findings and results will be described and discussed. Finally a conclusion of the topic and a perspective on future work is given.

[1] https://www.apple.com/ios/siri/
[2] http://www.google.com/landing/now/
[3] http://www.windowsphone.com/en-US/how-to/wp8/apps/meet-cortana

Chapter 2.

Background & Related Work

The thesis' concept for rule-based interaction is based on mobile interaction, multimodality and context-awareness. In this chapter a description and definitions of these areas' various aspects and an overview about related work will be given to provide a basis for the following chapters.

2.1. Mobile Interaction

Already in 2001 Dey et al. [11] noted that "computing devices and applications are now used beyond the desktop, in diverse environments, and [that] this trend toward ubiquitous computing is accelerating". The popularity of mobile devices (smartphones and tablets) has only proved this trend, so that human-computer interaction (HCI) nowadays increasingly deals with mobile interaction. The interaction with mobile devices can vary dramatically in comparison to the way persons typically interact with desktop computers. It is therefore useful to begin the description of the thesis' background with general characteristics of mobile interaction.

Due to the need for mobility and energy efficiency, smartphones have much smaller screens than most desktop systems, which can make it harder to conceive displayed information. Mobile devices also typically have less powerful hardware, which can slow down computational operations and with them the user when performing a particular action (e.g. looking for particular contents on a web page). These limitations can also result from slower connectivity (when compared to desktop computers which are mostly equipped with a fast Wi-Fi or Ethernet network connection).

Mobile interaction means interaction in steadily changing contexts. Attending to a constant flow of different events and stimuli and responding to those events with proper actions can result in cognitive limitations when interacting with a mobile device. We have to deal with various distractions and a multitude of events in direct our surrounding while the amount of attention a person can devote to the user interface is limited [4, 5]. Tamminen et al. [2] have observed 25 adult urbanites and their actions in everyday life to analyze interactions in mobile contexts. They observed that persons frequently experienced unplanned context changes and that these

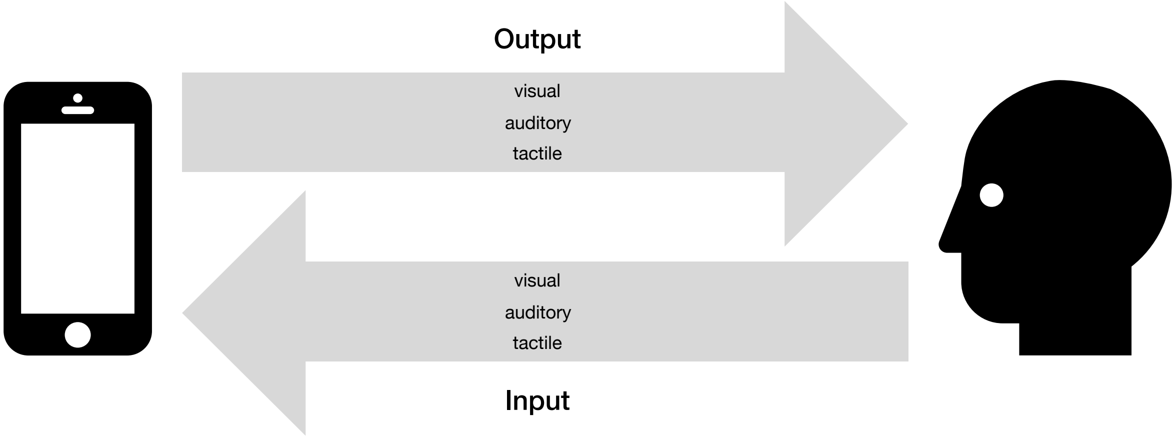


Figure 2.1.: Modalities described as in- and output communication channels in HCI.

"unplanned context changes lead to unplanned situational acts. [The] [...] participants often popped in somewhere or bumped into acquaintances on their way to their primary destination". Lemmelä et al. [5] furthermore point out that the mobile usage of devices requires suitable input techniques and that the social acceptance when interacting with mobile devices (sometimes resulting in impolite behavior) has to be considered when addressing mobile interaction.

2.2. Modalities & Multimodality

In human-computer interaction (HCI) a modality is a general type of communication channel which is defined according to the human senses (sight, hearing, touch, smell and taste). While the latter two senses do not play an important role in todays HCI systems (yet), modern smartphones can give output to us using the vision, audio and tactile modality and also receive input through these three communication channels (see Fig. 2.1). Modalities are realized by a series of in- and output hardware components built in modern smartphones. In the following these technical devices will be collected and described with regard to their properties, specific advantages and disadvantages, and their role for multimodal interaction.

2.2.1. Output Modalities

The **screen** is the smartphone's primary output regarding the visual modality. It can not only display graphics, animations and text while actively using the mobile device, but also display visual notifications when the smartphone is locked to show relevant information to the user. A screen has different properties such as size, brightness and colors, which can partly be adjusted. Advantages in comparison to other output modalities include the possibility to show highly specific information (compared for example to vibration, which can at most transfer information through

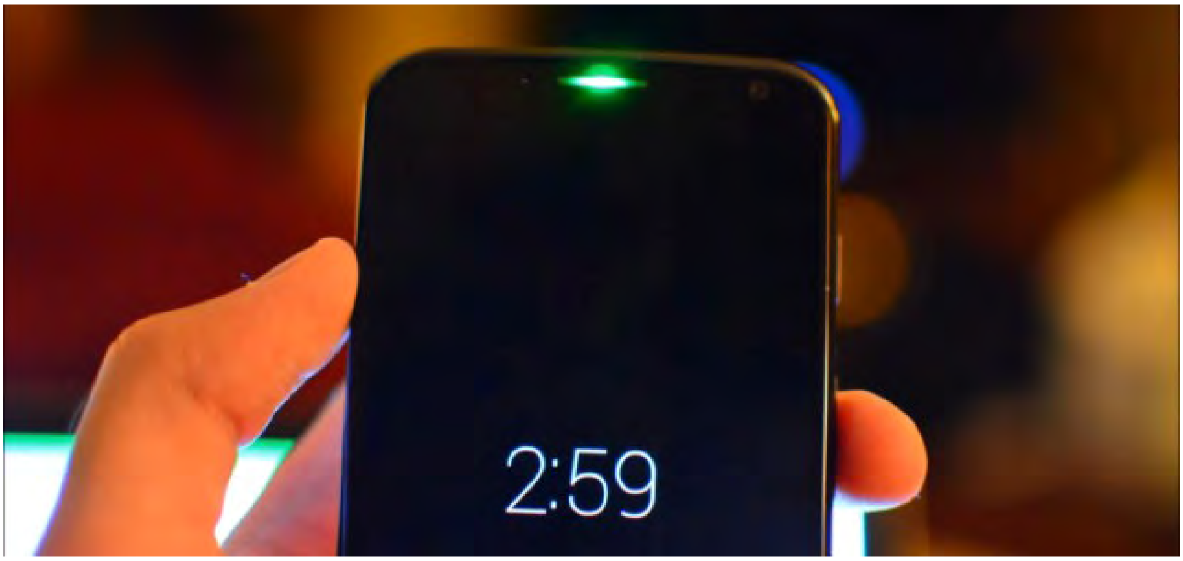


Figure 2.2.: Smartphone with a notification LED integrated in the earpiece area. Image from: [12]

different vibration patterns, intensity or amplitude) and the support of privacy [5]. The screen does however also have disadvantages as an output modality. It requires the user's direct focus: being out of sight it won't be noticed by the user. Also it can be challenging to read under extreme light situations, e.g. in bright sunlight.

The visual output modality is also realized as **LEDs** integrated in the smartphone's casing. Some smartphones feature an explicit LED to give the user visual feedback about notifications or other events (see Fig. 2.2), while other phones use the LED of the camera flash for the same purpose. Possible properties of the notification LED are color, intensity, frequency and rhythms or patterns. LEDs can be an unobtrusive way to notify the user or serve as an addition to other output modalities to make them more apparent. Just like the screen, LEDs require visual contact to be noticed by the user. Another disadvantage is that the conveyed information is relatively unspecific.

The auditory interaction channel is covered by the smartphone's **speakers** (typically one in the earpiece for calls and one in the bottom area of the casing for audio playback). They are used to give audio signals about incoming calls, notifications or other events to the user. With the advent of intelligent personal assistants like Apple's Siri, Google Now or Microsoft's Cortana speech output has become another important utilization. Possible properties of these sound signals are: volume, pitch, frequency, timbre, intensity and rhythm [5]. Sounds are advantageous when the user's focus is not on screen. They are rather obtrusive and draw attention. Obtrusiveness can obviously also be seen as an disadvantage under social circumstances. Also speaker output is problematic in noisy environments [5].

Lastly, **vibration** is used by mobile devices to communicate with the user through the haptic interaction channel. As vibration signals are also audible in many cases (for example if the smartphone is placed on a desk) vibration can also be seen as a part of the auditory modality. Possible properties are: intensity, amplitude, rhythm, and different vibration patterns [5]. Just like sound output, vibration is used to notify the user about calls, messages or other events. Another application is touch feedback ("vibrate on touch") when tapping buttons, e.g. on the (software) keyboard. Vibration signals can be a discreet form of interaction. They are very noticeable, even if the user's focus is not on the screen. However, tactile output can only convey a limited amount of information. Another disadvantage is that body contact is needed to perceive haptic feedback (except if the vibration is noticed aurally).

2.2.2. Input Modalities

Hardware buttons allow tactile input by the user. They are the oldest input method used on phones. While the number of hardware buttons constantly decreased with the advent of touch screens, most modern smartphones still have a small number buttons for elementary functionality (on/off/standby, volume). Even though hardware buttons might appear antiquated they have a number of advantages compared to other input modalities. Knowing about their shape and position on the device users can give input without focusing the screen. The position of a (bipolar) switch button can also give immediate feedback without actively interaction with the smartphone. Button input can be very fast, as it is not necessary to unlock the phone and to navigate to a particular view to evoke an action. Their restricted expressiveness and the limited size of mobile devices, however make the advantages of touch input (described in the next paragraph) clear.

The tactile input modality is also realized by **touchscreens**. Most devices use capacitive sensing, which takes human body capacitance as input. Advanced gesture recognition software interprets input touch data as commands and gestures. The integration of touch technology in small screens allows to present a large variety of (software) controls to the user and facilitates the usage of touch gestures. These so-called Post-WIMP (Windows, Icons, Menus, Pointer) interfaces were defined by Van Dam as an interfaces "containing at least one interaction technique not dependent on classical 2D widgets such as menus and icons" [13]. Multi-touch input employs familiar characteristics of reality such as the understanding of naïve physics which allows to make the interaction with a post-WIMP interface more like interacting with our everyday, non-digital world [14]. Disadvantages of the touchscreen as an input modality are the limited tactile feedback (sometimes realized with vibration signals which are however not selective, but effect the whole screen) and restricted perceptibility under extreme light conditions. It also requires the user to focus the screen, which can be problematic in some situations (e.g. when moving).

As mentioned earlier, intelligent personal assistants (like Apple's Siri, Google Now or Microsoft's Cortana) have become an important feature of modern smartphones. **Microphones** allow to receive auditory input in the form of speech commands. The underlying speech recognition technology has made it easy to perform simple tasks such as sending a text message, creating calendar events, setting alarms or finding nearby locations and the list of interpretable commands is steadily growing. Thus, one voice command can possibly replace a long series of touch inputs and navigation through different views. This fact can also be seen as the main advantage of the auditory input channel: speech can be a highly effective way of communication. At the same time speech is a very obtrusive form of giving input to a mobile device. Social concerns prevent many users from using their intelligent personal assistant in public. Another disadvantage lies in the imperfect reliability of speech recognition in comparison to other input methods. Especially in noisy environments, speech commands are often interpreted incorrectly, which can be very frustrating for users.

A mobile device's **cameras** (mostly one front- and one back-camera) are the only hardware components in the area of visual input modalities. Besides capturing photos or video, the majority of current smartphones does not feature visual input methods. One widely spread application are QR codes which can be scanned with the camera to quickly access information such as websites.

A summary of all in- and output devices classified by the used modality is given in Table 2.1.

Table 2.1.: Summary of in- and output devices of modern smartphones classified by used modalities.

Modality	**Output devices**	**Input devices**
Visual	Screen, LEDs	Cameras
Auditory	Speakers	Microphones
Tactile	Vibration motors	Hardware buttons, touchscreen

2.2.3. Multimodality

Simply put multimodality "refers to combination of multiple modalities" [15]. Williamson [16] has given a more exact definition of multimodal interfaces. The author describes them as "interfaces that specifically exploit the capabilities and affordances of more than one modality, either used together or separately as part of one interface". She furthermore emphasizes that "modalities, such as gesture, speech, visuals, audio, and vibrations, are still aspects of a multimodal interface even if they are investigated or used individually alongside other modalities.". Dumas et al. [9] identified properties and objectives of multimodal interfaces: they should "support and accommodate users' perceptual and communicative capabilities" and "integrate computational skills of computers in the real world, by offering more natural ways of interaction to humans".

The increasing number of in- and output technologies built in modern smartphones, which were described above, allow multimodal interaction. Outputs such as audio and vibration are used alongside each other and we use a variety of different input modalities to interact with a mobile device. The rule-based approach which will be introduced in this thesis aims to encourage multimodal interaction by supporting a suitable selection of output modalities in various contexts and enabling input methods, which combine different input modalities.

2.3. Context

The proposed rule-based approach for multimodal interaction uses *context* information to select suitable modalities. As the term *context* is relatively abstract, it is useful to look at a definition from the HCI field. Dey et al. [11] give an accurate definition of context as "any information that can be used to characterize the situation of entities (i.e., whether a person, place, or object) that are considered relevant to the interaction between a user and an application, including the user and the application themselves.".

How can context be determined technologically? Modern smartphones are not only equipped with an increasing amount of sensors, but also with several other hard- and software components which allow the collection of information characterizing the current usage situation. Based on the context feature space by Schmidt et al. [17] these smartphone components will be shortly described along two main categories of context information: *physical environment* and *human factors*, to gain an overview.

2.3.1. Physical Environment

Location

Current mobile devices use a mix of different technologies to determine the user's stationary or moving location at different levels of accuracy. **Global Navigation Satellite Systems** (GNSS) (e.g. GPS, GLONASS, Galileo or Compass) are mainly used for outdoor navigation with an accuracy of approximately 5-10 meters. The GNSS position calculation can be improved in terms of speed and reliability through assistance data provided over the cellular network (A-GPS). Especially in urban environments **Wi-Fi** access points are almost always available. Modern smartphones are capable of using large databases containing the location of wireless access points to determine their own location: once multiple Wi-Fi signals are available triangulation algorithms taking the signal strengths can be used to estimate the location with an accuracy of approximately 10-20 meters. If both, GNSS and Wi-Fi signals, are not available **Cell-Tower Triangulation** can be used as a fallback: based on the identifier of the currently used cell-tower mobile devices can determine their location with an accuracy of about 50-150 meters. Recently two additional wireless communication

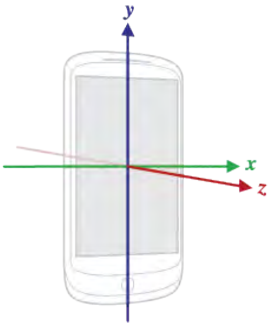


Figure 2.3.: Coordinate system (relative to a mobile device) that is used for the accelerometer, magnetometer and gyroscope sensors by the Android Sensor API. Image from: [20].

technologies which can be used for more exact localization were introduced: **Bluetooth Low Energy** (BLE, introduced in 2006) and **Near Field Communications** (NFC, introduced in 2004). BLE uses small (battery-powered) wireless transmitters (BLE Beacons) which continually transmit discovery signals over Bluetooth. If a smartphone is within the range of a BLE Beacon it can measure the signal strength to determine its distance (detectable distance: few centimeters to roughly 30 meters). Similarly NFC uses small wireless transmitters (called NFC Tags). Once a smartphone is within a short distance of roughly 4 centimeters a NFC Tag can use the transmitted radio waves to power-up an internal microprocessor, which can then transmit (location) data from its internal memory to the smartphone.

A combination of GPS and accelerometer signals can be used to determine a smartphone user's current **mode of transportation**. Advanced classification systems (e.g. [18]) can distinguish if a person is stationary, walking, running, biking, or in motorized travel with an accuracy of greater than 90%.

Environmental Conditions

Most smartphones are equipped with **accelerometer**, **magnetometer** and **gyroscope** sensors. An accelerometer returns acceleration measurements (in m/s^2) along the X, Y and Z axis and can be used to detect motion activities such as shake or tilt movements. A magnetometer measures the magnetic field (in μT) for all three physical axes and can be used in combination with an accelerometer to determine the direction with respect to the four cardinal directions. It could for example be used to display a compass. A gyroscope returns the angular velocities (in rad/sec) along each of the three axes and can be used to determine the correct orientation of a mobile device while in motion. It allows the detection of rotational movements like spinning or turning the device. Software sensor fusion methods using all three sensors collectively as created by Ayub et al. [19] allow to determine the current **orientation** (in degrees) in a coordinate system relative to the mobile device (see Fig. 2.3) even when the device is not moving.

Most mobile devices nowadays also feature an ambient **light sensor**. This sensor measures the ambient light level (in lx) and is frequently used to enable functionality such as controlling the screen brightness. Another frequently integrated sensing element is the **proximity sensor**, which allows to determine the proximity of an object to the screen of the smartphone (in *cm*). This sensor is often used to decide if a mobile device has currently been moved to a person's ear (when making a phone call).

Other specialized sensors which are built in a smaller number of smartphones make it possible to determine the **temperature**, ambient **air pressure** (barometer) or **humidity**. The data gathered through these sensors could for example help to determine the user's current altitude. Environmental conditions such as the current **weather** can be retrieved through web services.

A handheld's **microphones** and **cameras** can not only serve as user input (as described in Section 2.2), but possibly also be used as optical and auditory sensors. Microphones allow to determine the ambient loudness. More advanced audio processing could also analyze the type of background noise to identify particular situations (e.g. persons speaking). Cameras can gather visual information of the environment to determine the ambient light intensity or the type of light (sunlight, type of artificial light, etc). Using advanced processing algorithms could additionally allow to extract richer context information such as the detection of objects, landmarks, people or gestures [17].

While not strictly a sensor, a smartphone's **battery** possibly allows to draw conclusions about how long or intensively a mobile device has been used. Knowledge about the current battery level can help to extend the devices operating duration by reducing or disabling functionality with high power consumption.

2.3.2. Human Factors

User Activity

Apart from hardware components smartphones can nowadays also derive context information through various software applications and receive context-related data through an almost permanent connection to the world wide web. Calendar information could be used to determine a person's **habits** regarding recurring events. Certain events can reveal if a person is in a meeting, busy in other events or free at a given **time**. A person's **activity** and **emotional state** at the present time could be determined through the type of application or (web) content which is currently being used or consumed. If a user has been active in a reading application for a while this probably means that she or he is concentrated and unwilling to be disturbed at the moment, while the usage of social network applications might suggest a communicative mood.

Social Environment

Location-aware social network applications can nowadays gather information about **nearby persons**. On the condition that this information is made accessible to the mobile operating system (similar to calendar information), a high density of persons in the immediate environment (who revealed their current location to a social network) could for example indicate that a user is currently taking part in a public event.

Knowledge about the relationship a user has to other persons (belonging to defined circles of a social network) in close vicinity could furthermore allow to determine the **type of social interaction** in which she or he is currently involved. For example being surrounded by business contacts could mean that a person is currently in a formal meeting while being in a group of close friends indicates a relaxed situation. A summary of all described categories of context information is given in Table 2.2.

Table 2.2.: Overview about different categories of context information based on the context feature space by Schmidt et al. [17].

Physical Environment	Location	Rough Position
		Precise Position
		Mode of Transportation
	Conditions	Orientation
		Ambient Light
		Ambient Noise
		Proximity
		Time
		Battery
		Temperature, Air Pressure, Humidity
Human Factors	User Activity	Habits
		Emotional State
		Activity
	Social Environment	Nearby Persons
		Social Interaction

2.4. Related Work

The following section describes related work in the area of mobile context-aware systems and multimodal user interfaces. The focus is on context-aware systems, which use context information to modify their behavior without explicit user intervention.

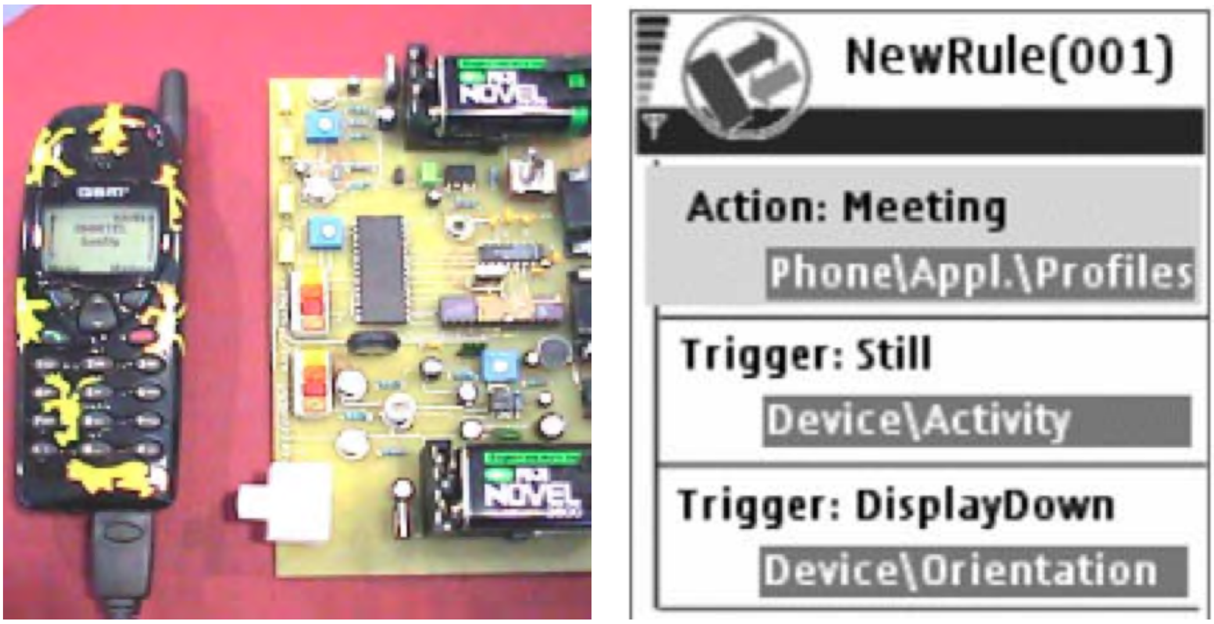
The development of context-aware frameworks and toolkits began with the emergence of mobile computing in the early nineties with the motivation to enable interactions which are suitable to different usage contexts. The *Active Badge System* developed by Want et al. [21] in 1992 consisted of wearable badges which wirelessly communicate to receivers in a building. This gathered context information was then used to direct telephone calls to the current position of the user.

Schilit et al. [22] first defined context-aware systems as systems that “adapt(s) according to the location of use, the collection of nearby people [...] [and] changes to such things over time”. They also described a rule-based system of simple IF-THEN rules to realize “context-triggered actions”. These actions allowed, for example to pop up messages on a handheld’s screen when different situations (e.g. entering a particular room) occur.

The *Context Toolkit* by Dey et al. [11] was one of the first toolkits providing software components which allowed developers to build distributed context-aware applications. These components, called *Context Widgets*, serve as context data sources aggregating context information from a variety of sources such as Active Badge devices (as described above), floor-embedded presence sensors or video image processing. One example is the *IdentityPresence Widget*, which is placed in a pre-specified location and then reports the arrival and departure of people at that location. Applications can subscribe to context widgets in order to receive context information. The main focus of applications targeted by the *Context Toolkit* was to display relevant context information to the user. Presented applications include an *In/Out Board*, which shows who is currently present in a building, and an *Information Display* displaying information relevant to a user’s location. At that time mobile devices were rather limited with regard to in- and output functions, so that the idea of adapting the device’s multimodal behavior did not yet play a major role.

Even before mobile devices were equipped with sensors themselves, Schmidt et al. [23] built a prototypical sensor board which could be connected to a Nokia phone still very limited in capabilities in contrast to a modern smartphone (see Fig. 2.4a). The prototypical setup allowed to automatically switch between predefined profiles of the phone. In an experiment the setup could for example recognize whether the user is currently outside and then automatically switch to the *Outside Profile*, which would increase the ringer volume and enable vibration. The work of Schmidt et al. has shown that it is feasible to recognize varying contexts using sensors and to adapt a mobile phone’s behavior accordingly.

Being unsatisfied with current commercial mobile phones and the cognitive load they impose on



NewRule(001)

Action: Meeting
Phone\Appl.\Profiles

Trigger: Still
Device\Activity

Trigger: DisplayDown
Device\Orientation

(a) Sensor board developed by Schmidt et al. [23] to equip a functionally limited Nokia phone with sensing capabilities.

(b) Context Studio by Korpipää et al. [24] allows to assign context-based triggers to different audio profiles in a graphical user interface.

Figure 2.4.: Sensor board developed by Schmidt et al. [23] and Context Studio by Korpipää et al. [24].

users to adjust modalities to their current mental state and context, Siewiorek et al [25] developed a similar system called *SenSay* which uses an external sensor box including accelerometers, light sensors and microphones attached to the user's hip. The gathered sensor data was analyzed and passed to a state machine which decided whether the user is currently uninterruptible, idle, active or in a normal state based on a set of predefined rules.

ContextPhone [26] is a a prototyping platform for context-aware mobile applications. Two of the main design goals were to provide context as a resource and the enhancement of existing smartphone applications (particularly messaging and calling functions). ContextPhone uses different sensors to retrieve context information, such as location (via GSM and GPS), activity information about the user (idle/active), communication behavior (calls, sent and received messages) and connectivity (surrounding Bluetooth devices, GSM availability). The platform was used to study users' mobility patterns by recording context information and to enhance existing phone applications with useful context information. "ContextContacts", for example, adds information such as the current location or the currently active audio profile of a contact to a phone's address book application).

While prior systems only allowed context-sensitive adaptations of the mobile devices in a hard-coded, predefined way that could not be controlled by the user, the possibility of running custom applications on off-the-shelf phones and more sophisticated screens soon allowed for user-controlled systems with a graphical user interface (GUI). Korpipää et al. [24] developed the *Context Studio* toolkit which allows users to bind contexts to application actions in a GUI. Selecting context-based triggers using a folder-subfolder navigation pattern (e.g. device → orientation → display down) and then assigning them to different audio profiles (e.g. meeting) gave the user the possibility to create simple, personalized rules to define their phone's behavior (see Fig. 2.4b).

Froehlich et al. [27] have developed a framework (*MyExperience*) for capturing both sensor- and human-based (interaction) data on mobile computing activities. The framework has an event-driven architecture of *Sensor*, *Trigger* and *Action* components. *Sensors* provide an abstraction for device state, user interaction, and environmental conditions. *Triggers* handle sensor data by defining the conditional logic that controls when to execute actions based on sensor states. And lastly, *Actions* are code snippets that are triggered based on sensor events. The framework's architecture allowed to perform actions such as displaying notifications, playing sound files, evoking vibration or LED flash patterns or launching external applications based on the sensor data and defined triggers.

AWARE by Ferreira [1] is a recently created comprehensive mobile context framework for Android. Similarly to the framework used in this thesis, it encapsulates implementation details of sensor data retrieval, does however not encompass the logic for evaluating context-based rules and evoking actions. The AWARE framework additionally allows to store context information on the mobile device and web servers, and provides an open-source library to researchers and application developers.

Recent commercial products using context information to evoke particular actions are the *Locale* [1] and *Llama* [2] Android applications. While offering a similar functionality as the application developed in this thesis, both applications are, however, not focusing on multimodal interaction, but rather on general automatic tasking such as en-/disabling Wi-Fi or Bluetooth connectivity or playing sounds if particular events take place (see Fig. 2.5).

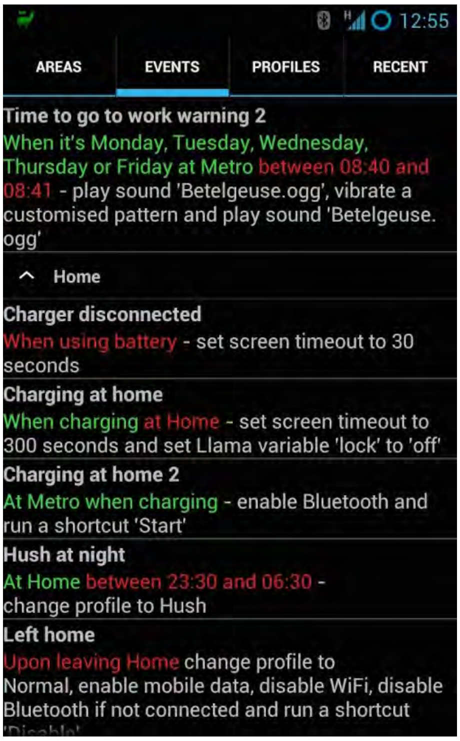
Interesting approaches to include non-conventional input modalities in mobile interaction have been presented by Möller et al. [28]. Interaction techniques such as *pointing* (aiming at an object with the smartphone), *touching* (identifying an object by bringing the phone close to it) or *scanning* (pointing at a visual tag with a camera) make use of a mobile device's sensors and optical technologies and have shown to be well appreciated and effective for users.

Computer vision has recently gained increasing attention as a possibility to acquire context information in the direct surrounding of the user. Tools which support simulating the interaction between vision-based localization and the user interface have been presented in the area of indoor navigation [29]. Google's *Project Tango* [3] uses an Android phone equipped with highly special-

[1] http://www.twofortyfouram.com

[2] https://play.google.com/store/apps/details?id=com.kebab.Llama

[3] https://www.google.com/atap/projecttango/



Time to go to work warning 2
When it's Monday, Tuesday, Wednesday, Thursday or Friday at Metro between 08:40 and 08:41 - play sound 'Betelgeuse.ogg', vibrate a customised pattern and play sound 'Betelgeuse.ogg'

^ Home

Charger disconnected
When using battery - set screen timeout to 30 seconds

Charging at home
When charging at Home - set screen timeout to 300 seconds and set Llama variable 'lock' to 'off'

Charging at home 2
At Metro when charging - enable Bluetooth and run a shortcut 'Start'

Hush at night
At Home between 23:30 and 06:30 - change profile to Hush

Left home
Upon leaving Home change profile to Normal, enable mobile data, disable WiFi, disable Bluetooth if not connected and run a shortcut 'Disable'

Figure 2.5.: Location-aware mobile application *LLama*.

ized sensor hard- and software to track the full three-dimensional motion of a mobile device. The research goal is to approach human sensory abilities by creating “a mobile device that shares our sense of space and movement, that understands and perceives the world the same way we do” [30].

Chapter 3.

Concept

The following chapter describes the concept behind the developed application. After positioning our rule-based approach in comparison to alternative approaches such as manual modality switching and fully automated systems, the general goals and a formal definition of our concept will be described. The concept phase was guided and accompanied by a focus group to gain insights about the usage of different modalities, to generate ideas and to discuss user interface possibilities. The main part of this chapter is a description of the resulting user interfaces, which gives an overview of the developed application and explains the rationale behind our design decisions.

3.1. Rule-based Multimodal Interaction

Our general aim was to support smartphone users in selecting suitable modalities by utilizing context information from the environment, thus improving the interaction with the mobile device and reducing the user's task load.

The current situation for smartphone owners requires the user to manually switch between modalities. Whenever the user's environment changes in a way that requires a change in out- or input methods (e.g. entering a library), the user manually triggers modality changes (e.g. by disabling sounds via a button or in the mobile operating system's settings). As illustrated in Fig. 3.1 (A) this approach results in a high task load for the user, while the smartphone does not react autonomously in any way. It is questionable if the diversity of different situations we encounter throughout a day can still be coped with through a series of on/off switches (e.g. sound on/off, vibration on/off) in a user-friendly way.

An extreme approach in the other direction could be seen in a fully automated system without a graphical user interface (GUI) which learns from the user's behavior in differing situations and autonomously applies modality changes after going through decision-making algorithms. For example, it is conceivable that a system observes a user frequently muting her or his phone between 22 and 23pm in the evening and then automatically performs this task for the user after

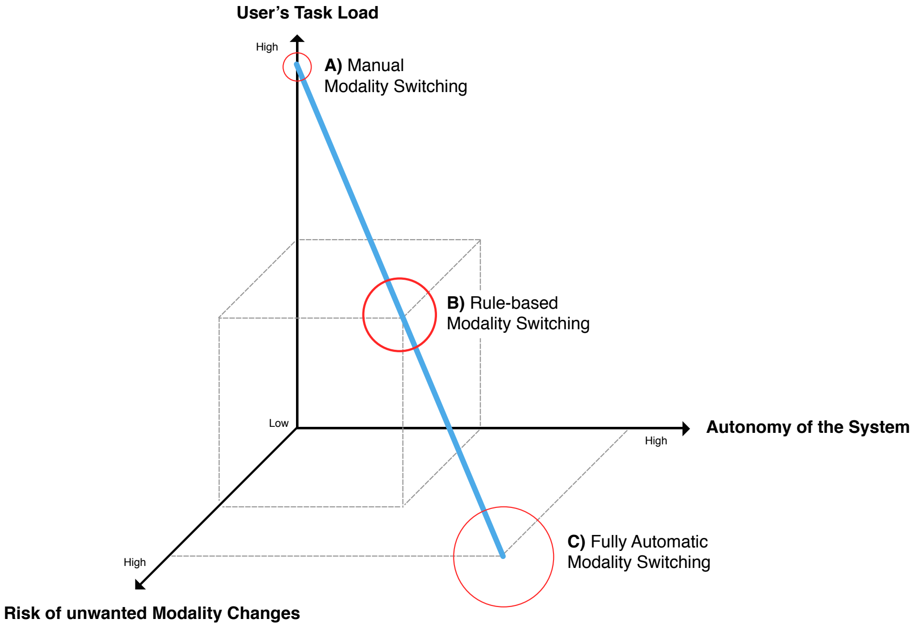

User's Task Load
High
A) Manual Modality Switching
B) Rule-based Modality Switching
Low
Autonomy of the System
High
High
C) Fully Automatic Modality Switching
Risk of unwanted Modality Changes

Figure 3.1.: Illustration of the relation between a user's task load, the autonomy of a system and the risk of unwanted modality changes for context-aware modality switching.

a certain learning period without. This approach would not require any inputs or tasks at all by the user and is characterized by a high level of system autonomy, however the risk of unwanted modality changes is high, as the user has no possibility to control the system's behavior (see Fig. 3.1, C). In this case it is questionable whether too much autonomy on the side of the system could at times result in unwanted modality changes combined with insufficient transparency for the user ("black box" behavior): a user might sometimes not be able to comprehend why the mobile device's settings have been changed.

3.1.1. Goals

On the background of these observations the following goals (besides providing suitable modalities) for a system supporting multimodal interaction emerged:

Reduced Task Load

A system for rule-based interaction should reduce the user's task load by automating predictable tasks. While the creation of rules might require some user input (mostly in the initial setup stage) the system should take work out of the user's hands by applying rules, defined for frequently occurring situations and resulting in modality changes, automatically. Such a rule-based system would be positioned between the previous two approaches with respect to user task load, system autonomy and risk of unwanted modality changes (see Fig. 3.1, B).

Control & Discoverability

Lemmelä et al. [5] noted that "even the best modality configuration is worthless if the user is not able to discover the interaction possibilities". Users might not be able to comprehend and keep track of untransparent automatic changes, if there is no possibility to realize why they occurred. A system should therefore provide a central GUI to leave control and an overview about possible modality changes to the user. Allowing the users to define and edit rules by themselves could make modality configurations discoverable.

Consideration of Human Behavior

Unlike computers humans behave nondeterministically. Only little changes in our environment can make us deviate from normal behavioral patterns. For example, the time we go to bed can vary depending on where we are and which (unexpected) events we encounter throughout a day. On the other hand it is hard to imagine conditions under which we would not want to get obtrusive feedback (using sounds and vibration) when, for example, being on the bike. While rule-based interaction can still bear the risk of applying modality changes when the user does not want them by way of exception, it should allow the user to cover only those situation which are experienced in a comprehensible and regular manner. A system should not try to cover every aspect of human behavior by applying modality changes automatically in every situation.

3.1.2. Formalization

Our approach to rule-based multimodal interaction allows the user to define *rules* for multimodal interaction with the smartphone. The following two rule examples illustrate this idea:

- **Example A**: *If I am at the library (context), mute the phone (output modality).*
- **Example B**: *If I hold the phone in front of me and press the volume button (input modalities), open the camera application (action).*

After conceiving different rules for *in*- and *output* modalities, it became clear that conceptual differences and differences regarding the technical possibilities exist:

Output modalities can be adapted to changing contexts in a meaningful way (consider Example A). A rule can adjust particular output modalities for the duration of a particular context and then change them again when the context is left or another context appears.

Input modalities on the contrary typically evoke singular actions, such as opening a smartphone application (see Example B). While it is useful to have different input modalities for the same task to offer suitable input methods in varying contexts, it is generally not necessary to disable certain input modalities in a particular situation. The user can choose the appropriate input modality by her- or himself when it makes sense in a particular context. It is for example not necessary to disable speech input in a particular situation. If it is not a suitable input method in that situation, the smartphone owner can just use text input or any other suitable input method.

Still, for both, in- and output modalities, the definition of rules appears as a promising way to support multimodality. As described more formally in the following, our rule-based system allows: (1) the selection of suitable output modalities in changing contexts, and (2) the definition of new input methods, which combine different input modalities.

For use case (1) rule-based multimodal interaction consists of

- **context factors** given through environmental conditions or human factors (e.g. location or ambient light) on the one hand, and
- **output modalities** being adapted accordingly (e.g. disabling sound) on the other hand.

For use case (2) rule-based multimodal interaction consists of

- **input modalities** realized as input methods like pressing a button or performing a motion gesture on the one hand, and
- **actions** (e.g. opening an application) on the other hand.

3.2. Focus Group

To gain insights about the usage of different modalities, to generate ideas and to discuss user interface possibilities a focus group was held in the concept phase. The six participants consisted of four research assistants (all with a Diploma degree) and two students (both with a Bachelor of Science as their highest degree) from the computer science field. The group consisted of five males and one female with an average age of 27.3 years (standard deviation: 2.0 years). The currently owned smartphones were two LG Nexus 4, one LG Nexus 5, one Samsung Galaxy SII, one Apple iPhone 4S and one iPhone 5. On a scale from 1 to 5 three participants estimated their smartphone expertise with 4, whereas the other three estimated it a with 5. The focus group began with a brief introduction to the topic and took 1 hour and 30 minutes. The generated ideas and the requirements we derived from the findings are summarized in the following.

Diverse Usage of Modalities

Modalities were used in different ways by each participant. Whereas one participant used vibration and the smartphone's notification light as primary output modalities and avoided speech input, another participant stated that he did not use vibration at all (as he did not sense it properly), but made heavy use of speech input as an efficient method to give short commands to the phone. A third participant in turn reported that he relied solely on vibration signals, while the fourth participant kept his phone completely silent (no audio signals, no vibration) almost all the time and preferred visual display notifications (having his smartphone mostly placed on the desk next to him). These findings suggest that user interfaces for rule-based multimodal interaction should be flexible enough to handle a diverse set of individual preferences.

Standard Settings

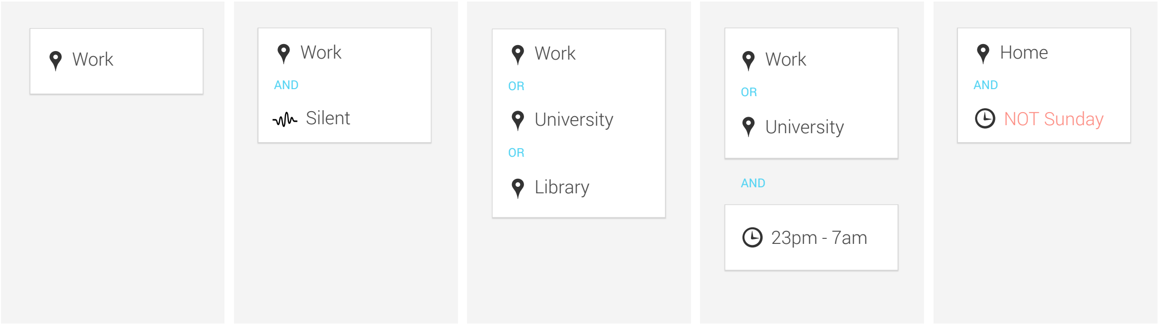
All participants reported that they changed both input and output modalities rarely. On the input side touchscreen and hardware buttons were used primarily, while speech and motion gesture interaction was reported to be problematic for most participants. Mentioned reasons included inconsistent implementations of motion gestures across different apps and varying quality and reliability of the phone's speech recognition. On the output side all participants chose an unobtrusive setting with concentration on visual and haptic feedback to reduce disturbances by the smartphone. One participant also mentioned that he often forgot to revoke activated output modalities which frequently resulted in an unwanted setting for him. All in all the participants mainly used one standard setting as a "lowest common denominator" for varying situations. On this background ruled-based modality switching appears as an opportunity to support the user in utilizing a broader spectrum of modalities without additional operational effort.

Profiles

Automatic modality changes based on the current user context were welcomed by all participants on the condition that they work reliably. At the same time the request for (situation) profiles as known from some older feature phones was expressed, so that taken together a rule-based approach for modality changes with a profile-like description for each rule appears to be useful.

Awareness

Active rules and the resulting modality settings should be displayed in a subtle way, but still be visible at first glance according to the participants. Another requirement which all participants consented on was that the system should always leave full control to the user ("I am afraid to lose control over my settings."). A practical suggestion in this regard was to display active rules in the smartphone's notification area. Furthermore a possibility to actively disable automatic settings or to overwrite them through conventional means like the physical volume buttons was requested. Automatic changes could then be reactivated when the next rule becomes effective.



(a) One location context factor.

(b) One location and one ambient noise context factor.

(c) Three location context factors.

(d) More complex: two location context factors and one time context factor.

(e) NOT statement.

Figure 3.2.: Examples of common situations as representations in the GUI.

3.3. User Interfaces

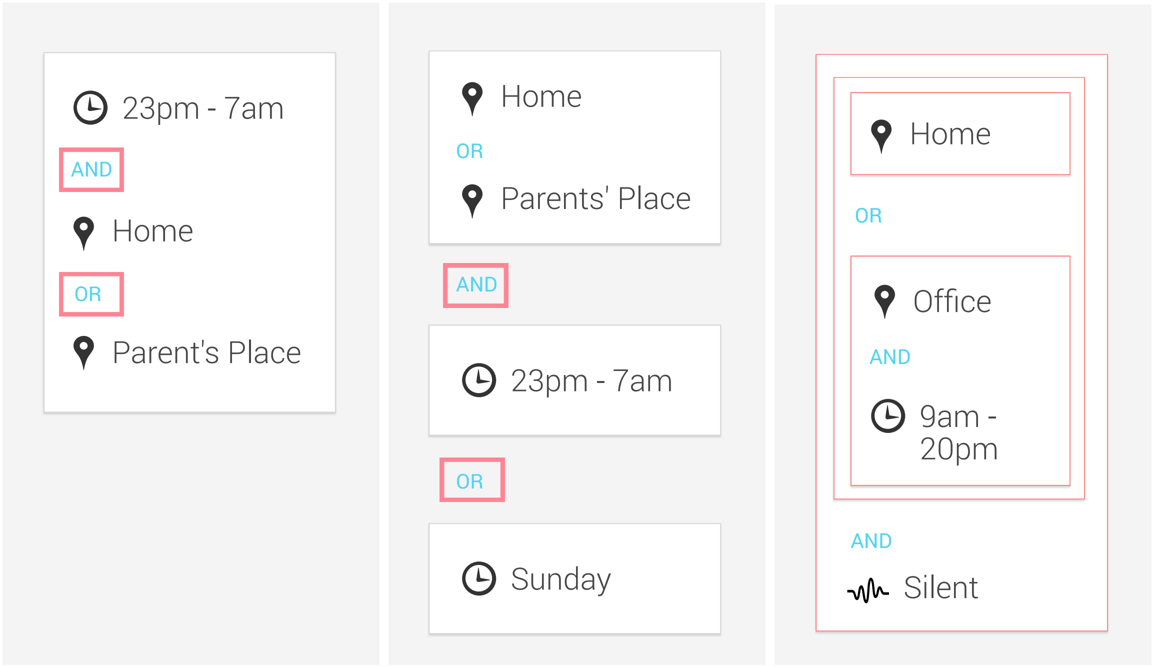
The following section shows the user interfaces which were created on the background of the findings and derived requirements. The section gives an overview over the developed GUIs and explains the rationale behind our design decisions.

3.3.1. Representation and Definition of Situations

Situations can be described as a combination of different context factors. A user could, for example, define the following situations: "I am at work and it is silent around me" or "I am at the library or at university". These combinations, which are logical expressions from a mathematical point of view, can also become more complex. A possible example could be: "I am at library or at university and the time is between 8am and 17pm". The first step was to think about ways to represent and define such situations in the GUI.

Common Situations

It is important to optimize a GUI for the most common scenarios. Confronting the user with a multitude of different GUI elements needed only in very rare cases can be confusing and make the application unusable for beginners and intermediate users. We therefore started by identifying situations which are presumably most common (see Fig. 3.2). The assumption that most created rules are rather simple was later confirmed by the results of a field study we conducted: most rules created by the participants had a simple structure with an average of 1.15 selected context factors (see Fig. 5.27b).



(a) Example of an ambiguous expression.

(b) Example of ambiguity between groups.

(c) Example of nesting.

Figure 3.3.: Situation structures and layouts to be avoided.

Representations to be Avoided

One requirement was that it should be possible to describe complex situations with the help of logical expressions. Not all users are, however, familiar with details about boolean operators (mainly: *AND*, *OR*, *NOT*) and how they are evaluated. We therefore also thought about situation structures that should be avoided so that users are not confronted with unwanted outcomes.

Fig. 3.3a shows an example which could be ambiguous from a user's perspective. For the shown expression, the user would probably expect that the last two location context factors are evaluated together against the first (time) context factor, because they are both location context factors. This, however, is not the case as *AND* operators are evaluated before *OR* operators. This problem can also occur between groups (representing parentheses which control the order of operations in an expression) if three or more groups are possible (see Fig. 3.3b).

Nesting would be an alternative to groups when creating more complex expressions. However, readability and interpretability may suffer strongly, especially when nesting expressions deeper than one level (see Fig. 3.3c). We therefore used groups to represent more complex situations in the final GUI.

Defining an Expression

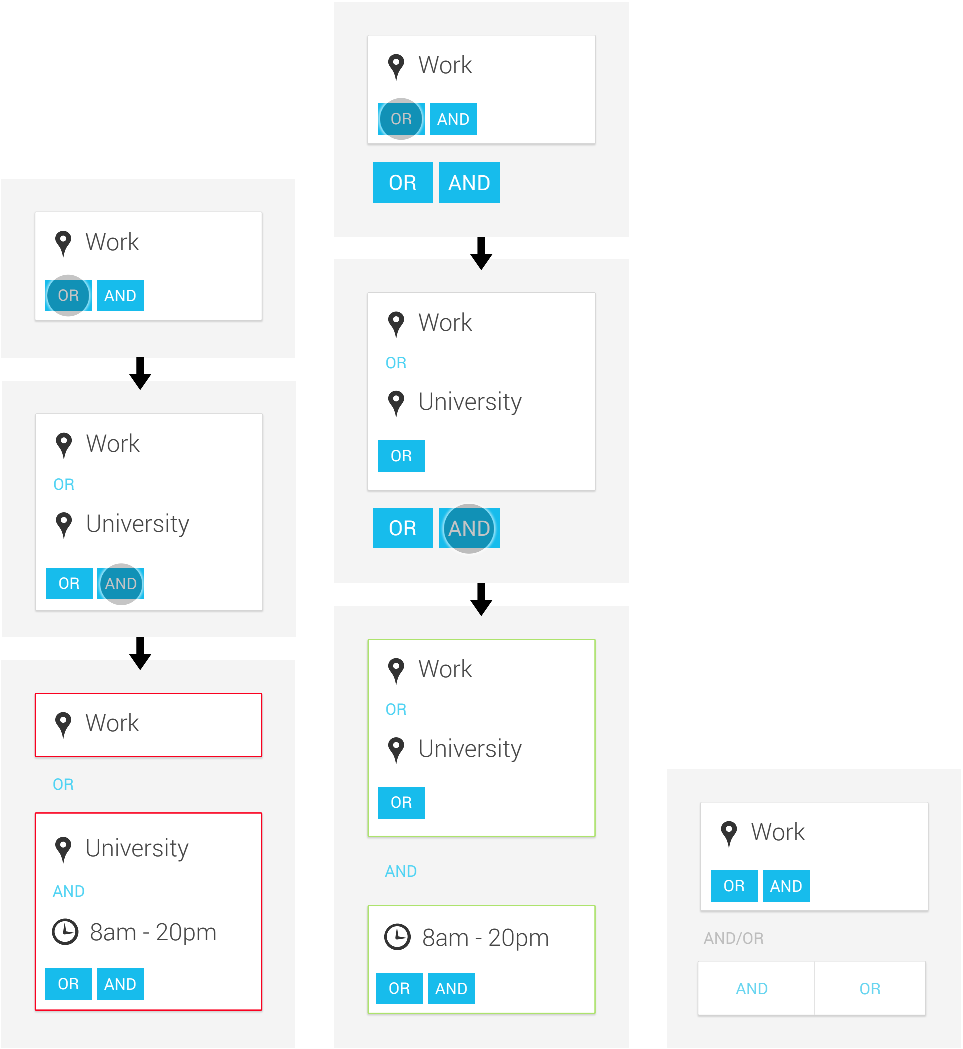
On the background of the previous findings, the last step before continuing with following parts of the GUI was to determine the best way for users to connect different context factors when creating an expression.

One approach to avoid ambiguity concerning the evaluation of logical expressions would be to automatically create a group if an *AND* connection follows an *OR* connection (see Fig. 3.4). This does, however, not permit the creation of structures as shown in Fig. 3.2d ([Work *OR* University] *AND* [23pm-7am]). Supposed the user has already added the two location context factors, adding the time context factor with an *AND* would create a new group containing the university context factor and the time context factor, which is not what the user intended. While an equivalent situation could be defined by beginning with the time context factor, the user should not be patronized in how he defines a situation. Without having knowledge about the evaluation of boolean expressions, the user might furthermore not understand or foresee why such automatic changes take place.

The approach we chose for the final GUI design is shown in Fig. 3.4b and 3.4c. It allows the user to either add statements within a given group or to add a new group (second row of buttons). This allows to create different situation structures more freely. The creation of the situation shown in Fig. 3.2d is now possible. Ambiguity is prevented by firstly, setting a boolean operator type (*AND*/*OR*) for each group after the first statement was added within that group, and secondly, by limiting the number of possible groups to two.

One problem with with this approach was that the user has to choose between four visually similar buttons already at the entry point. This appears problematic when considering that the most common situations are relatively simple ones as shown in Fig. 3.2a - 3.2c. The addition shown in Fig. 3.4c reduces visual complexity by displaying the second row of buttons less prominently and hinting to a second group. Thus the call to action is laid on the more frequently needed *OR* and *AND* buttons at the top.

This approach introduces some limitations: it does not allow to define any desired logical expression (with an arbitrary nesting depth). It was not our aim to allow the definition of every conceivable rule. Instead our concept aims for a trade-off between the possible complexity of a situation on the one hand and a user friendly, unambiguous creation process of rules on the other hand. Our concept focuses on more common simple situations, but still allows relatively complex expressions by offering groups.



(a) Approach 1, automatic group creation.

(b) Approach 2, manual group creation.

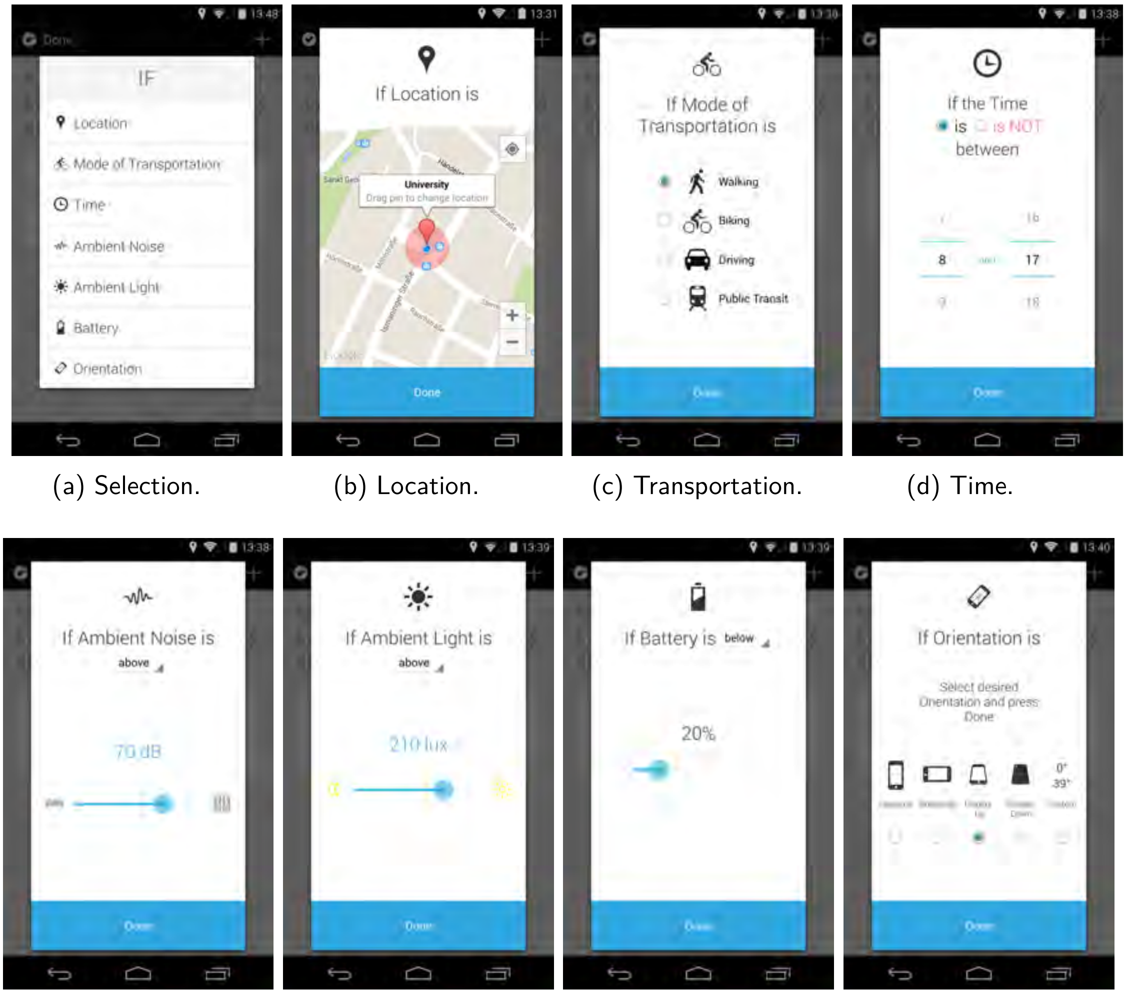
(c) Approach 2.1, addition to Approach 2.

Figure 3.4.: Different approaches to creating an expression.

3.3.2. Context Dialogs

After designing how context factors can be combined to let the user define situations, the next step was to find concepts for defining context factor parameters, for example: *location* (context factor) is *university* (parameter).

Fig. 3.5a (Selection) shows a menu which is presented to the user to select a context factor. The location dialog (see Fig. 3.5b) allows to set a location (and a description) with a certain radius (red circle) as a parameter. The target pin is initially set to the user's current location as this is presumably a common input value. The view shown in Fig. 3.5c allows to select one of four modes of transportation: walking, biking, driving or public transit. The next context dialog (see

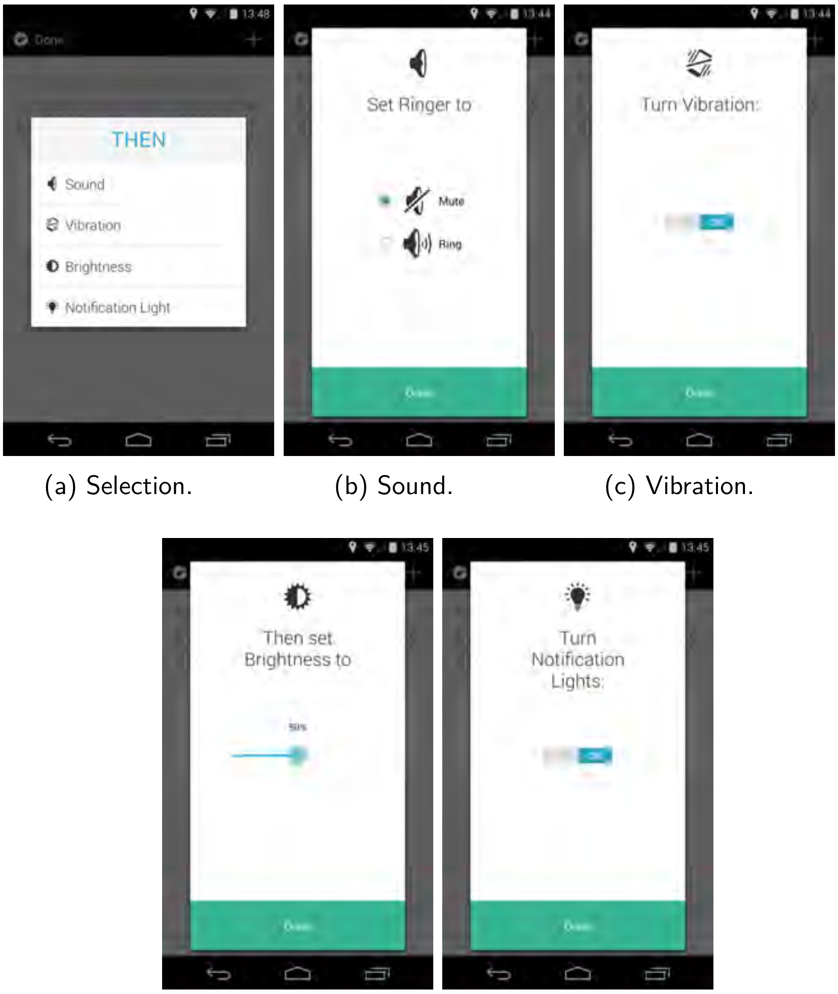


(a) Selection. (b) Location. (c) Transportation. (d) Time.

(e) Ambient Noise. (f) Ambient Light. (g) Battery. (h) Orientation.

Figure 3.5.: GUIs created to define context factors.

Fig. 3.5d) shows two picker controls to define a time interval (e.g. between 8am and 17pm). The time dialog additionally features the possibility to create a *NOT* statement. This option is deliberately not shown in other context dialogs, as more intuitive ways to express negations exist here: it is for example more intuitive to define the ambient noise parameter as *below 60dB* than to specify it as *NOT above 60dB*. Both the ambient light (see Fig. 3.5f) and ambient noise dialog (see Fig. 3.5e) show a selection field (*above* or *below*) and a slider control to specify the desired value. A vertical bar behind the slider control indicates the current noise (or light) level to give the user an orientation. The battery context dialog (shown in Fig. 3.5g) was designed in a similar way. The orientation dialog (see Fig. 3.5h) presents commonly used orientation parameters (*upwards*, *upwards*, *display up*, *display down*) as shortcut options, but also allows to set a custom orientation.



(a) Selection. (b) Sound. (c) Vibration.

(d) Screen Brightness. (e) Notification LED.

Figure 3.6.: GUIs created to define output modality switches.

3.3.3. Output Modality Dialogs

Similar dialogs were created to define parameters for different output modalities. The menu shown in Fig. 3.6a allows to select a auditory (sound/ringer), visual (screen brightness and notification LED) or tactile (vibration) modality. Using the sound, vibration and notification LED dialogs (see Fig. 3.6b, 3.6c and 3.6e) a user can control whether each setting should be turned on or off. The screen brightness dialog (see Fig. 3.6d) allows to specify a percentage parameter using a slider control.

3.3.4. Icon Language

The application uses icons to clarify the meaning of GUI elements (see Fig. 3.7). We developed an icon language containing more than 30 icons. While most common icons are easy to interpret it can be challenging to find a suitable representation for more abstract notions such as orientation. It is also important to design icons for similar actions in a distinguishable way (for example the first icon represents ambient light while the second stands for screen brightness).

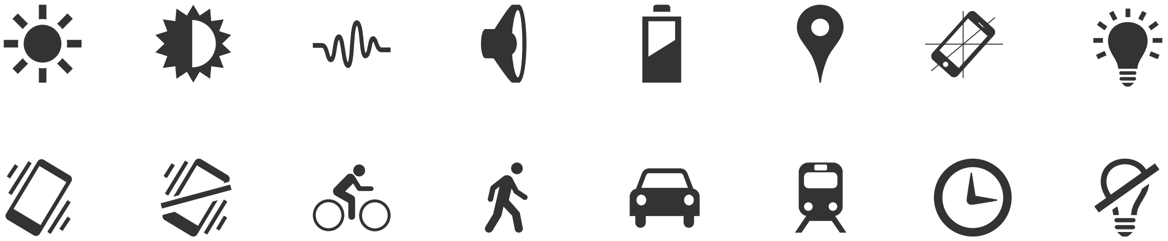


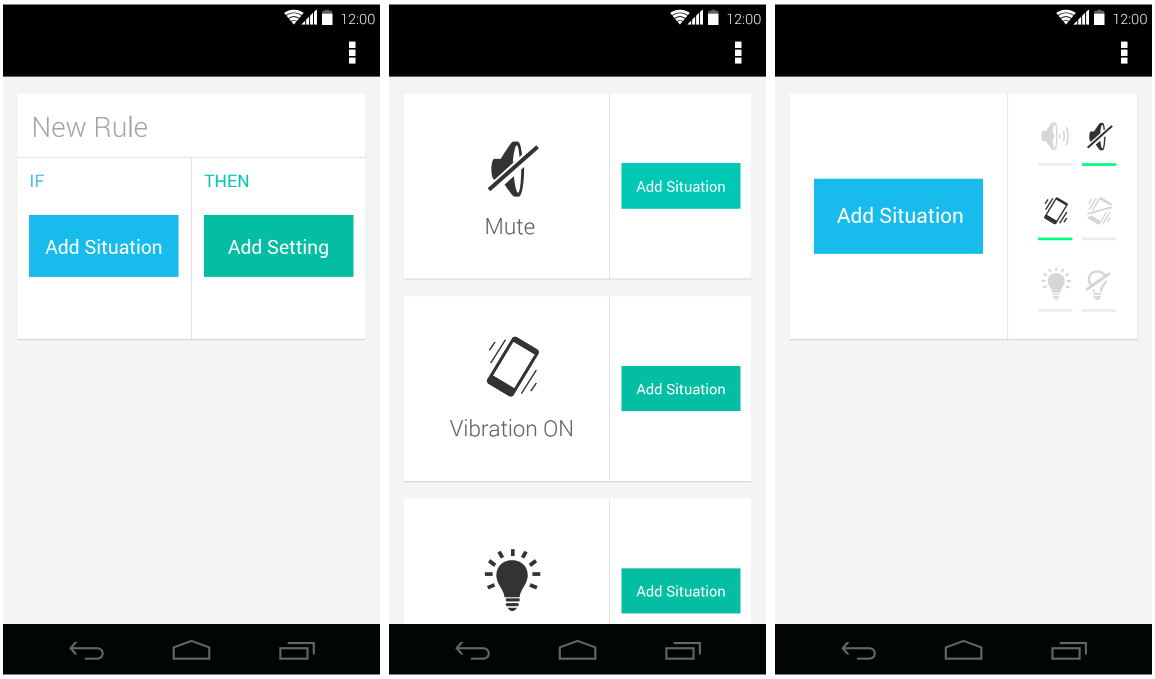
Figure 3.7.: Selection of icons which were used to clarify the meaning of GUI elements.

3.3.5. Rule Layouts

After designing the process of defining situations (as simple logical expressions) and the views for specifying context factors and modalities (as dialogs), the next step was to develop a layout to present a complete rule.

Our first approach displayed context factors on the left side and modalities in the right side in a visually *balanced* layout (see Fig. 3.8a). The user can equally add context factors or modalities on either side until the desired rule has been built. We used the terms "Situation" and "Setting" instead of "Context" and "Modality"' throughout the GUI, because the latter two terms are rather abstract and can thus be harder to understand for users.

One interesting question that emerged during the concept phase, was whether persons think about context-modality combinations in either a *situation-oriented* or a *modality-oriented* way. An orientation towards modalities would mean that users first think about different modality zones such as "loud" or "silent" and then assign different suitable situations to these zones. A similar approach was for example chosen in the *Context Studio* application by Korpipää et al. [24] where context-based triggers are assigned to different audio profiles (see Fig. 2.4b). The *modality-oriented* layout realizing this mind set is shown in Fig. 3.8b. Here, the user could, for example, assign the situation "location is library" to the modality zones "Mute" and "Vibration ON". An orientation towards situations on the other hand would mean that users primarily think about particular situations in which they would like certain modality changes to be applied. The corresponding *situation-oriented* layout is shown in Fig. 3.8c. Here users will most likely begin by adding a situation on the left side and then select suitable modalities by activating switches on the right side.



(a) *Balanced* rule layout. (b) *Modality-oriented* rule layout. (c) *Situation-oriented* rule layout.

Figure 3.8.: Rule layout variants corresponding to different user mind sets which emerged during the concept phase.

3.3.6. Navigation Flow

The navigation flow of the application is mostly based on dialogs (for context factors and modalities, as described above). Once evoked they will appear on top of the underlying rule view. Thus the user is presented an own view for each task without completely leaving the current context of the main view (which is always visible beneath the dialog). Fig. 3.9 shows the views a user navigates to create the rule: “If I am at university, mute the phone”.

3.3.7. Awareness

There are different ways to provide awareness on modality changes to the user. As our first approach we designed a *widget* (a small application which can placed on the smartphone’s home screen, see Fig. 3.11a or 3.11b), which dynamically updates its content whenever the user’s context changes. If, for example, the user enters the location of his workplace (and has defined a rule for this context), the widget would display this situation and the assigned modality changes. A widget is a rather unobtrusive method to provide awareness, as it does not interrupt the user’s

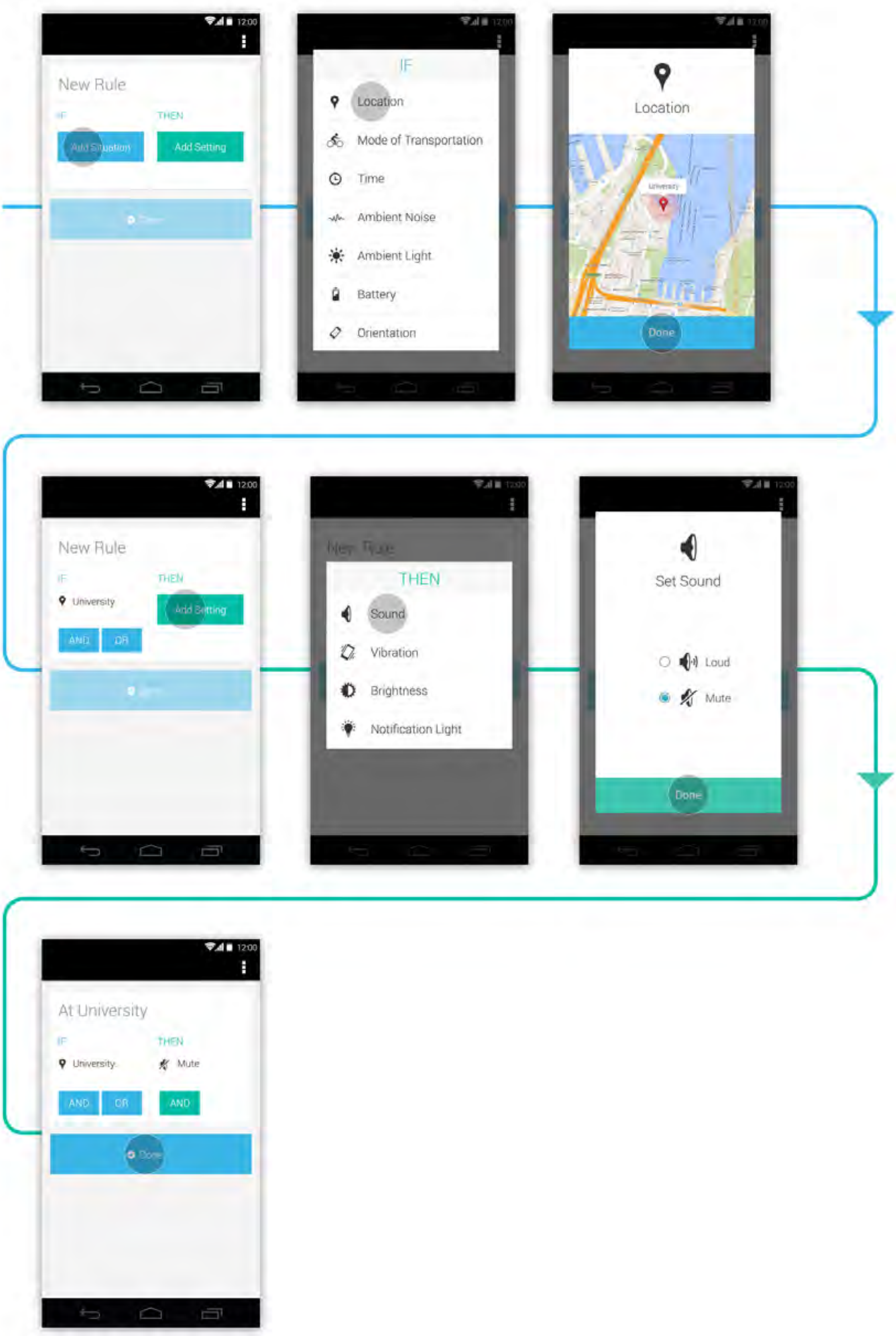


Figure 3.9.: Navigation flow when creating a rule (using the *Balanced* layout).

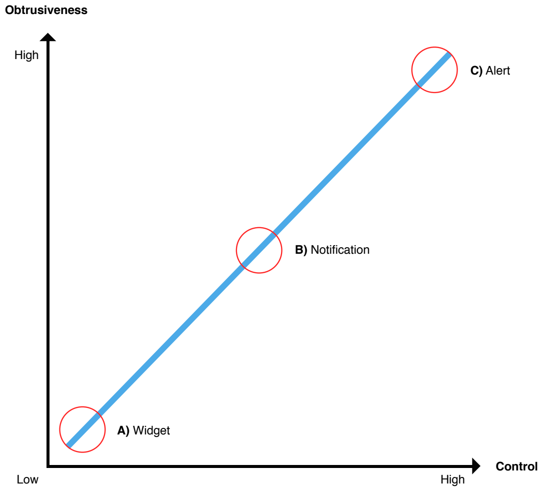


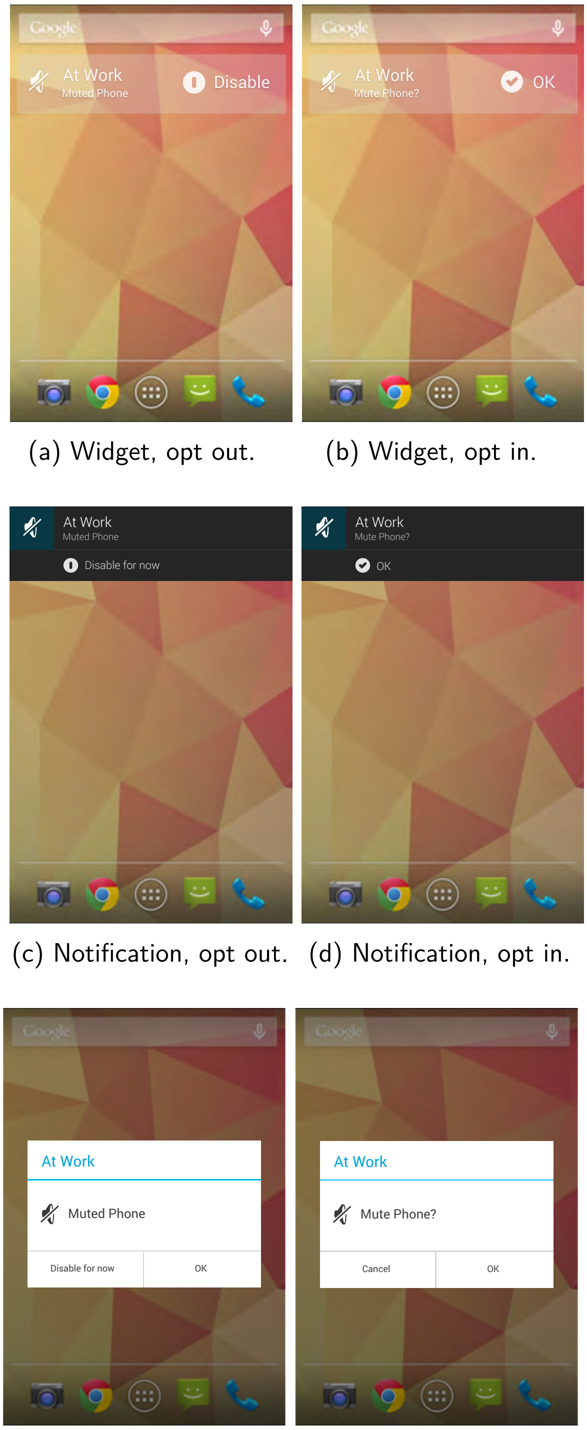
Figure 3.10.: Three awareness methods: *widget*, *notification* and *alert*, positioned in comparison to each other regarding obtrusiveness and control.

work flow. On the other hand, a person will only notice modality changes when actively looking at the widget, which gives her or him less control over the system.

On this background we thought about other methods, which differ from each other along these two characteristics (obtrusiveness and control). A modal *alert* is very obtrusive, as it is displayed on the screen in a prominent way and demands an immediate response by the user (see Fig. 3.11e or 3.11f). On the other hand an alert also offers immediate control about modality changes.

Most mobile operating systems allow to show *notifications* to the user. Notifications can be seen as a middle course between widgets and alerts regarding obtrusiveness and control: they are typically displayed in the status bar area in the top area of the smartphone screen and are automatically dismissed by the system after a short duration (see Fig. 3.11c and 3.11d). While not forcing the user to interact they allow user input if desired. Diagram 3.10 illustrates how these three awareness methods can be positioned in comparison to each other regarding obtrusiveness and control.

Regarding the interaction with each of these awareness methods two possibilities emerged: a) once a rule becomes effective the system can either explicitly ask the user before changing modalities: *opt in*, e.g. "Mute the phone?" or b) just apply the modality changes but offer the possibility to disable a modality for a given situation: *opt out*, e.g. "Muted the phone. (Disable for once?)". An overview of all different methods is given in Table 3.11. The designs in the left column show each method with the interaction type *opt out*, the right column demonstrates the *opt in* possibility. Especially for users who have a high level of trust in automated system actions or are very reluctant to interruptions by the smartphone, another possibility would be to not display modality changes at all. We later investigated all seven possibilities in a laboratory study to gain insights about the users' preferences.



(a) Widget, opt out.

(b) Widget, opt in.

(c) Notification, opt out.

(d) Notification, opt in.

(e) Alert, opt out.

(f) Alert, opt in.

Figure 3.11.: Overview about different methods to provide awareness about modality changes to the user.

3.3.8. Rule Creation Alternatives

Besides creating rules by manually adding context factors and modalities, we additionally thought about alternative ways to create a useful set of rules.

The idea of the alternative *Suggestion* is based on a machine learning approach: while a person is using her or his smartphone the app could register if certain modalities are frequently used in particular situations (a user might for example frequently mute his phone at the library) and then present according rules to the user. If a suggested rule appears to be helpful the user can add it to her or his set of rules. Suggested rules may also be modified by the user in hindsight if desired. The designed GUI for this rule creation alternative can be seen in Fig. 3.12a.

The second alternative we conceptualized (*Snapshot*) presents a pre-built rule containing a complete set of context factors and all currently selected modalities to the user. Instead of manually adding desired situations and modalities the user can remove unwanted parts until the desired rule is reached. This approach appears especially helpful if a person adjusts the smartphone's modalities while actually being in a particular situation and wants to take a "snapshot" of that condition. The GUI for this alternative is shown in Fig. 3.12b.

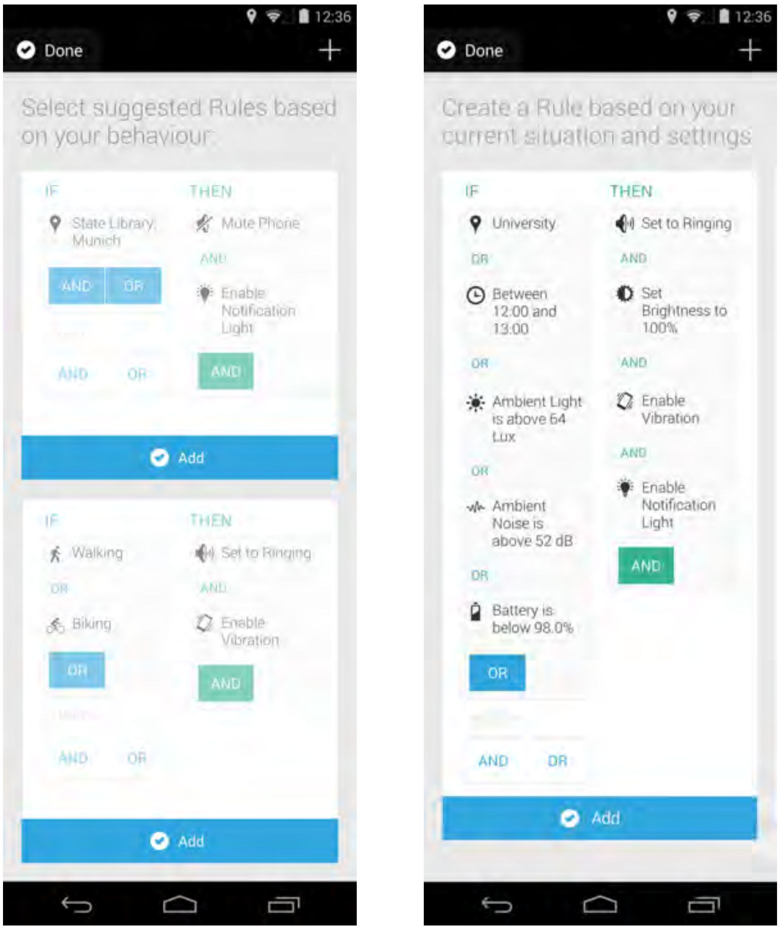
3.3.9. Input Rules

As described in Section 3.1.2 we see conceptual differences in the context-aware usage of output and input modalities. In contrast to output modalities which can be adjusted to different contexts in a meaningful way, suitable input modalities can be actively chosen by the user when it makes sense in a particular situation. It is generally not necessary to disable certain input modalities in a particular situation. Still the definition of rules appears as a promising way to support multimodality on the input side.

Similarly to the context dialogs we designed to define contextual situations, we therefore created GUIs, which enable the user to define rules combining different input modalities to evoke actions such as opening an application (see Fig. 3.13e) on a mobile device. Fig. 3.13a shows the menu which allows to select one of four offered input modalities. The GUIs allow the user to add button-press actions to a rule (see Fig. 3.13b), or to record touch- (see Fig. 3.13c) or motion gestures (see Fig. 3.13d).

Mimicry Gestures

Research on the topic of multimodal input methods has shown that providing additional input modalities such as motion gestures can be used to simplify interaction with mobile devices [31].



Select suggested Rules based on your behaviour:

IF
State Library, Munich
AND OR
AND OR

THEN
Mute Phone
AND
Enable Notification Light
AND

IF
Walking
OR
Biking
OR
AND OR

THEN
Set to Ringing
AND
Enable Vibration
AND

Create a Rule based on your current situation and settings

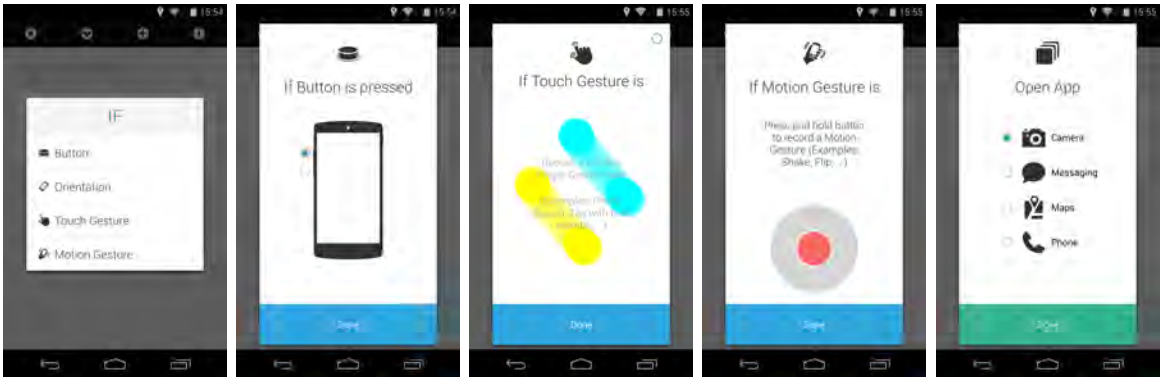
IF
University
OR
Between 12:00 and 13:00
OR
Ambient Light is above 64 Lux
OR
Ambient Noise is above 52 dB
OR
Battery is below 98.0%
OR
AND OR

THEN
Set to Ringing
AND
Set Brightness to 100%
AND
Enable Vibration
AND
Enable Notification Light
AND

(a) Rule creation alternative *Suggestion*

(b) Rule creation alternative *Snapshot*

Figure 3.12.: Rule creation alternatives: *Suggestion* & *Snapshot*.



Press and hold button to record a Motion Gesture (Examples: Shake, Flip, ...)

(a) Input modality selection. (b) Hardware button dialog. (c) Touch gesture dialog. (d) Motion gesture dialog. (e) App trigger dialog.

Figure 3.13.: GUIs created for the defintion of input rules.

On the other hand, experiments have revealed two major problems regarding the usage of non-conventional input methods: 1) Rico and Brewster [32] explored the **social acceptance** of participants performing different gestures in public places. They discovered that participants disliked gestures when they were uncommon and noticeable by nearby persons. 2) In another experiment Lemmelä et al. [5] evaluated different input modalities in particular contexts such as walking or in the car. They noticed that "even the best modality configuration is worthless if the user is not able to discover the interaction possibilities". The participants were not able to to discover possible input gestures without the moderator's help, so **discoverability** appears to be another problem.

The rule-based approach bears the possibility to overcome the problem of poor discoverability by providing a central GUI where interaction possibilities are always visible as input rules. Additionally the defined input rules might be memorable as the user defines them by her- or himself. On the background of social acceptance we came up with the idea of *mimicry gestures*, which mimic natural behavior of people using their smartphone to trigger common tasks. Imagine for example someone taking a photo. It will most likely look similar to what we see in Fig. 3.14.

Based on frequently observable postures of people interacting with a smartphone the following *mimicry gestures* emerged:

- Hold your phone in front of you and press the volume button to open the camera application.
- Tap the screen with both thumbs in an alternating way to open the messaging application.
- Make a spread & pinch gestures on the screen to open the maps application.
- Move the phone to the ear and back to open the phone application.

These gestures, described as input rules in the GUI, are shown in Fig. 3.15. Besides being socially acceptable through their hardly noticeable appeal, we see possible advantages in these

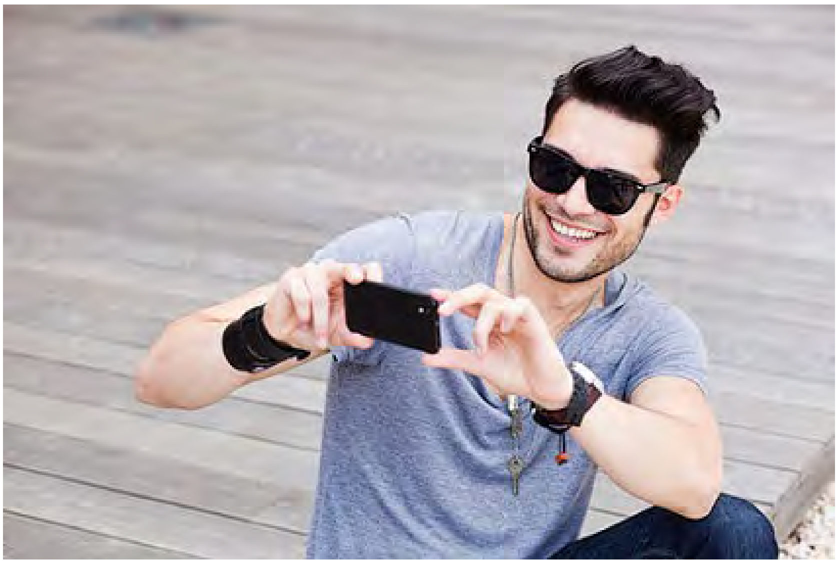
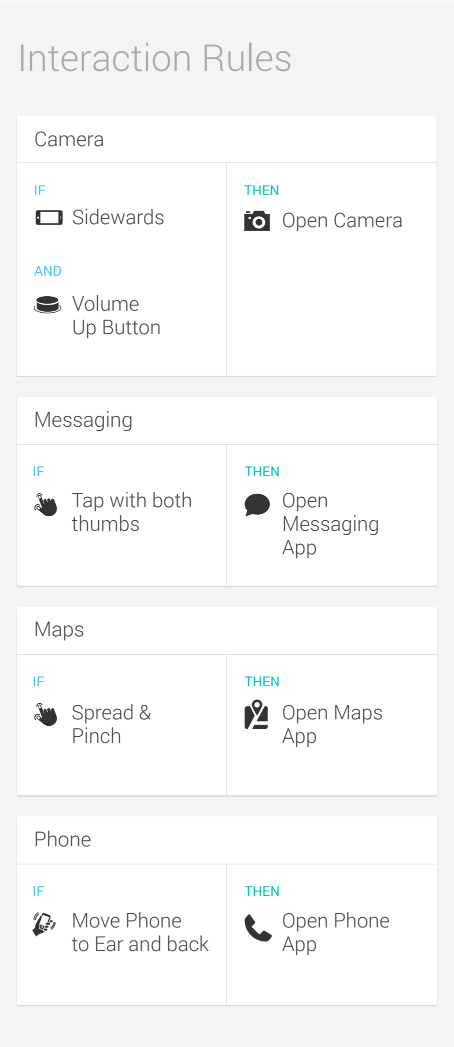


Figure 3.14.: Typical posture when taking a photo (image from http://www.huawei-university.at).

input methods in mentally preparing the user for the task to be evoked and the fluid transition to the actual task. For example, if a user performs the camera gesture, she or he will be in the posture for actually taking a photo even before the camera application has been started. While input rules using the touch screen should only be evocable from defined areas such as the lock or home screen, motion gestures can generally be used in every situation without conflicting with common tasks performed on a mobile device.

A well performing recognition of touch and motion gestures requires advanced processing and interpretation of the given input data, which was not in the thematic scope of the thesis. Still we wanted to investigate the idea of input rules and the idea of mimicry gestures as an addition to the rule-based approach, so that we evaluated these in a Wizard of Oz experiment as part of a laboratory study. The results, together with an evaluation of all demonstrated GUIs will be described in Chapter 5. The next chapter describes the implementation of the developed Android application.



Interaction Rules

Camera

IF Sidewards

AND Volume Up Button

THEN Open Camera

Messaging

IF Tap with both thumbs

THEN Open Messaging App

Maps

IF Spread & Pinch

THEN Open Maps App

Phone

IF Move Phone to Ear and back

THEN Open Phone App

Figure 3.15.: Conceptualized input rules as defined in the GUI.

Chapter 4.

Implementation

The following chapter describes the *M3I Framework for Context-Based Mobile Multimodal Interaction* which builds the basis of the developed application and components which were added to the framework to extend its functionality, as well as the developed Android application including user interfaces and underlying background services. The implementation was realized in Android Studio [1] using an LG Nexus 4 smartphone running Android 4.4 (KitKat, API level 19) as a test device.

4.1. Framework

The application is built on top of the *M3I Framework for Context-Based Mobile Multimodal Interaction* developed by Möller et al. [33]. The *M3I* framework simplifies the access to context information, e.g. ambient light level, device orientation or battery information, allows to control different modalities, e.g. sound or vibration, and wires together the logic behind context-sensitive modality switches based on rules.

4.1.1. Description

The following description of the *M3I* framework is based on its documentation [33] and briefly describe its structure and components to provide an understanding for the following parts of the implementation.

Context Factors and Context Groups

Context factors represent different types of context information, e.g. the ambient light level, the time, or the battery level of the device. A `ContextFactor` can be created by defining

[1] http://developer.android.com/sdk/installing/studio.html

a ContextGroup it belongs to (e.g. OrientationContext or LightContext), and a context method that provides its value. The following example demonstrates the usage of a Float context factor which describes the device's battery:

```
FloatContextFactor battLevel =
  new FloatContextFactor(
    new BatteryContext(this),
    BatteryContext.FLOAT_GET_BATTERY_LEVEL);
```

Triggers

Triggers allow to perform actions like modality switches and can be evoked as a consequence of context changes. A trigger could for example change the devices's screen brightness or disable sound output.

Rule-Based Wiring

Rules wire a LogicalExpression (based on Boolean algebra which can be evaluated as true or false depending on the values of the covered ContextFactors) with desired Triggers.

Logical Expressions

A simple form of a logical expression is given through a Statement containing a ContextFactor and an Operator (e.g. equals() or greaterThan()). BinaryExpressions allow to create more complex logical expressions using Boolean operators such as AND, OR and NOT. The following example shows a statement which expresses that the battery level is above 50% and a binary expression containing two statements:

```
Statement isAboveHalfCharged = new Statement(
  battLevel, FloatOperator.greaterThan(50f));

BinaryExpression exp = new BinaryExpression(
  BinaryExpression.EXPRESSION_OR,
  isAboveHalfCharged, isPluggedIn);
```

Rule Evaluation

The framework contains an Evaluator, which evaluates rules in a defined update interval and executes triggers. The following example demonstrates the activation of the Evaluator in an

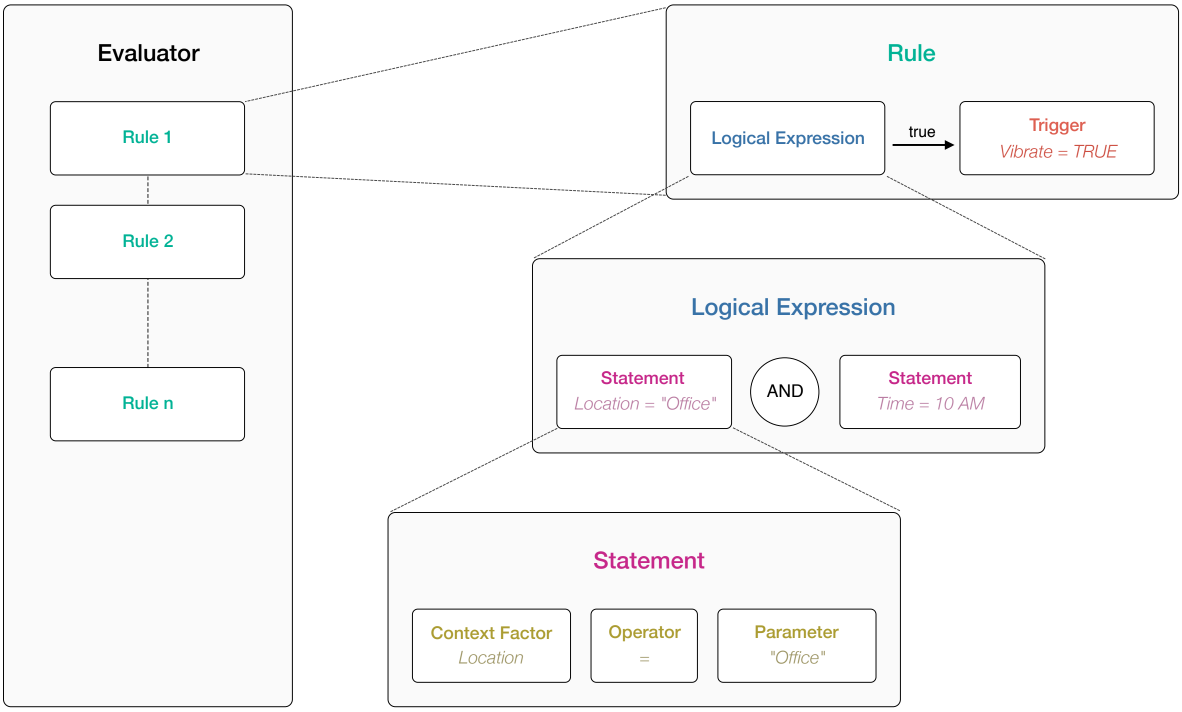


Figure 4.1.: General structure and components of the *M3I Framework for Context-Based Mobile Multimodal Interaction*, which provides the basis for the Android application's implementation. Drawing based on [33].

Android application (using a 1000 milliseconds update interval):

```
Evaluator e = new Evaluator(1000);
e.addRule(rule);
e.start();
```

Fig. 4.1 provides an overview of all explained components and the framework's general structure.

4.1.2. Extending the Framework

The *M3I* framework was extended with additional context groups, factors, operators and triggers to cover a wider range of context information and modalities, as well as functionality regarding rules and the evaluator to offer more flexibility in different usage scenarios.

Context Groups, Factors & Operators

A `LocationContext` group which presents the user's location as contextual information was added to the framework. The class makes use of system location services, which allow to obtain periodic updates of the mobile device's geographical location. Developers can define a geo-fence radius (e.g. 50 meters), which is used by the evalutator to evaluate whether a location statement is true. Location coordinates are typically defined as LatLng (latitude, longitude) objects, so that the framework was also extended with a LatLngContextFactor and a LatLngOperator. A context factor of the type LatLng indicating the device's current location and a statement comparing the user location with another location can be defined as follows:

```
LatLngContextFactor userCoordinates = new LatLngContextFactor(
  IContextGroup.CONTEXT_LOCATION,
  LocationContext.LATLNG_GET_LOCATION
);

LatLng libraryCoordinates = new LatLng(<latitude>, <longitude>);

Statement userIsAtLibrary = new Statement(
  userCoordinates, LatLngOperator.equals(libraryCoordinates)
);
```

Additionally a NoiseContext group which represents the ambient noise level in the user's environment was added. It uses short audio recordings received from the mobile device's microphones to determine the current noise level in decibel (dB). A context factor of the type Double indicating the current ambient noise level and a statement comparing it with a target noise level can be defined as follows:

```
DoubleContextFactor noiseLevel = new DoubleContextFactor(
  IContextGroup.CONTEXT_NOISE,
  NoiseContext.DOUBLE_GET_NOISE_LEVEL
);

Statement louderThan70dB = new Statement(
  noiseLevel, DoubleOperator.greaterThan(70);
);
```

Triggers

A BrightnessTrigger which changes the screen brightness, a VibrationTrigger which turns vibration on or off, a NotificationLightTrigger which turns the notification LED on or off and an AppTrigger which launches designated Android applications (e.g. the camera application) were added. The following example demonstrates a trigger changing the screen brightness:

```
BrightnessTrigger bt = new BrightnessTrigger();
```

```
bt.setAction(BrightnessTrigger.ACTION_SET_BRIGHTNESS);
bt.setBrightness(255);
bt.trigger(); // -> sets screen brightness to 100%
```

Evaluator

The initial version of the *M3I* framework's `Evaluator` did not yet consider different rule states and could only handle one rule at at time. The framework's `Evaluator` was extended with the possibility to handle a set of rules and to keep a record of different states for each rule (`active/inactive`, `enabled/disabled`).

Maintaining different states for each rule is necessary on the following background: a rule clearly defines how modalities are changed when a user enters a defined context. There are, however, different possibilities to react when the user leaves a context again. One possibility would be to just keep the rule's modality settings applied even when it is not valid anymore, only a consequent rule would then change modalities the next time. A second possibility would be to apply a previously defined set of default modality settings (e.g. sound ringing, vibration on, screen brightness 100%, notification LED off). Another possibility would be to automatically return to the previously selected modality setting (e.g. if the smartphone is initially set to silent and the owner enters and then leaves a loud environment where the sound is set to ringing, the smartphone would automatically return to the previous silent setting).

On the assumption that it would be the behavior expected by most users the functionality to automatically return to previous modality settings was implemented (the results of the conducted laboratory study later showed that this behavior was indeed preferred by the majority of all participants, see Section 5.1.6). Adding an `active/inactive` state to each rule, allowed to implement this behavior in the following way:

- Each rule is continually evaluated in the defined update interval.
- When a rule is first evaluated as true, the currently active modality settings are saved as the rule's `elseTriggers` and the rule is set to `active`.
- When the rule is afterwards first evaluated as false, the recorded `elseTriggers` are executed (thus restoring the previous modality settings) and the rule is set to `inactive`.

Another idea which emerged during the focus group was the possibility to temporarily `disable` rules. A user might for example disable certain rules referring to work situations when being on vacation. This functionality was implemented as a simple `enabled/disabled` state for each rule. In the extended version of the `Evaluator` only `enabled` rules will be evaluated. The new version of the `Evaluator` extended with the described functionality is illustrated in Fig 4.2.

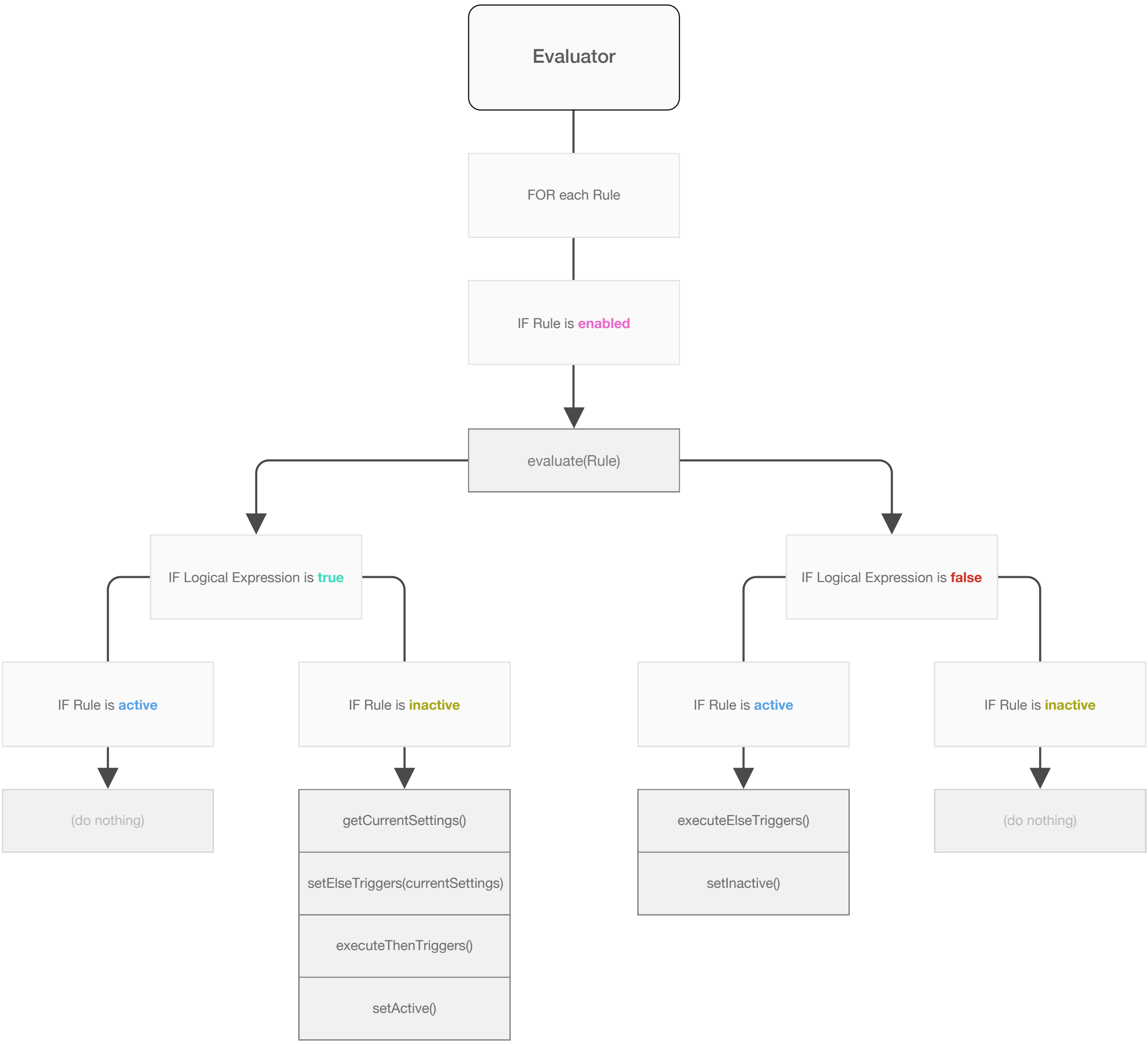


Figure 4.2.: Extended version of the `Evaluator`. Maintaining `active/inactive` states for each rule and saving current modality settings as `elseTriggers` allows to return to the previous modality settings once a rule is first evaluated `false`. An `enabled/disabled` state allows to temporarily disable certain rules.

Serialization

The early version of the *M3I* framework did not yet allow object persistence. In the extended version persisting `Rule` objects was realized using Java Serialization. To achieve this the `Rule` class and all subsequent classes (`LogicalExpression`, `Trigger` etc.) implement the `java.io.Serializable` interface. Rules can be persisted across application sessions using the `saveRules()` method of the `Evaluator` and restored using its `loadRules()` method.

4.2. Android Application

Android applications are designed around the Model-View-Controller (MVC) software architecture. The behavior and appearance of the GUI is defined by *views* which are presented to the user and respond to user input and *controllers* which pass data to and from the views and determine the navigation flow between them.

In Android, controllers are subclasses of `Activity` or `Fragment`. `View` is the base class for interactive UI components (such as buttons or text fields) and layouts (containers defined in a XML layout file that hold other views). The main controllers and views defining the GUI of the developed Android application are described in the following.

4.2.1. User Interfaces

RuleListActivity

The `RuleListActivity` is the main activity and is presented to the user when starting the application (see leftmost screen in Fig. 4.3). It extends the Android `ListActivity` class and shows an overview of all rules by binding to the `RuleStore` as the data source. As a `ListActivity`, the class also exposes event handlers for selecting a rule from the list. The individual rows in the list view are defined in the `RuleAdapter` class (extends the Android `ArrayAdapter` class), which inflates the layout for each row. Each row has an activity indicator in the top right corner which shows if a rule is currently active.

From here the user can start the `RuleActivity` by tapping the "+ (Add Rule)" menu item, which will create a new rule, or by selecting an existing rule item in the list, which will present the `RuleActivity` with the selected rule.

RuleActivity

The `RuleActivity` encapsulates all functions which are needed to create a new rule or to show, edit or delete existing rules (see Fig. 4.4). Besides adding a profile-like title, this activity allows the user to define a situation by connecting different statements (each consisting of a context factor with a parameter) using boolean operators (AND/OR) and groups (see Fig. 4.4b). Desired modalities for a situation can be added as triggers. When creating a rule the activity dynamically checks whether all necessary components have been added by the user. If so, the done button will be enabled (it is previously disabled) and the complete rule may be added to the user's set of rules.

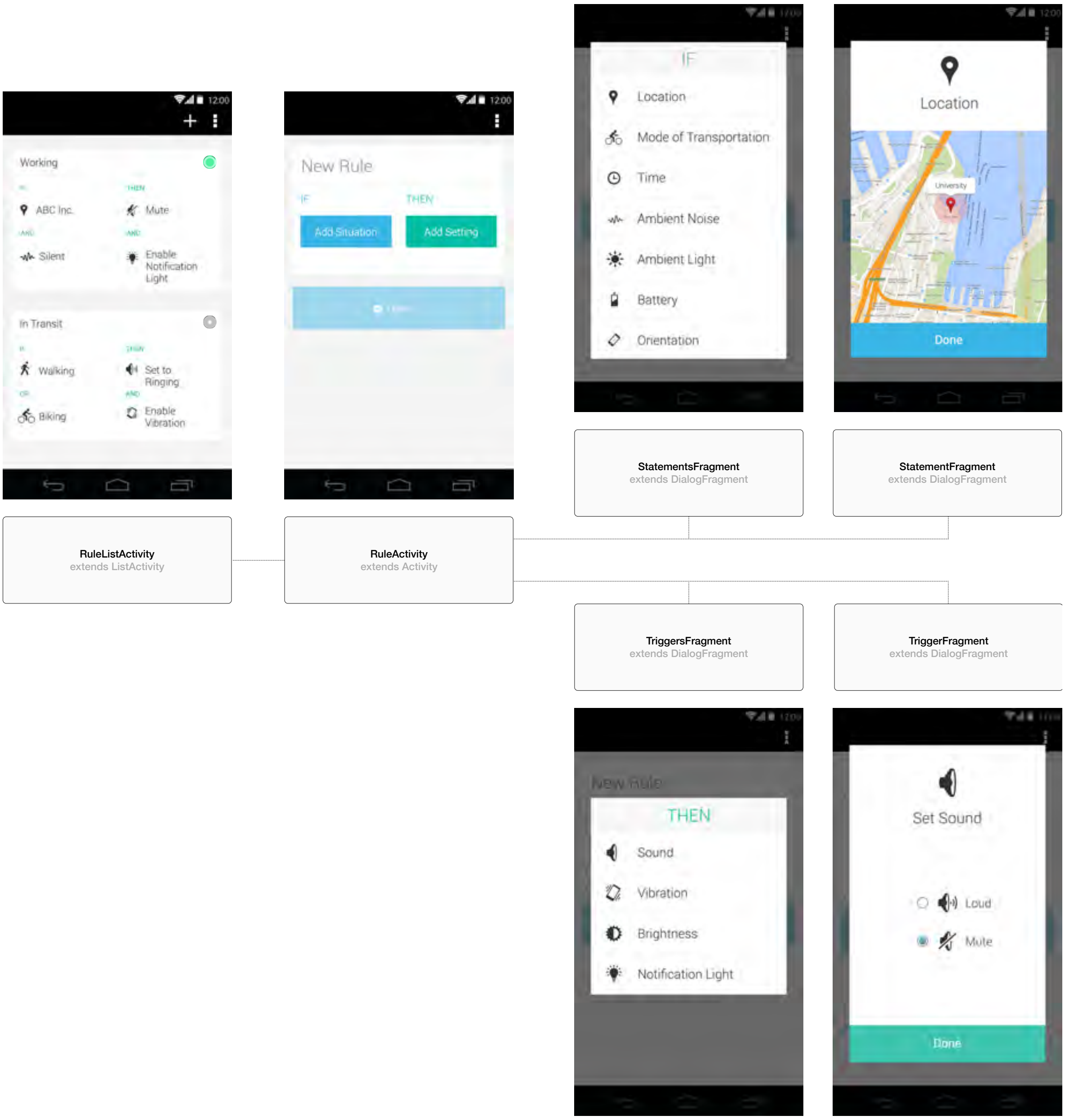
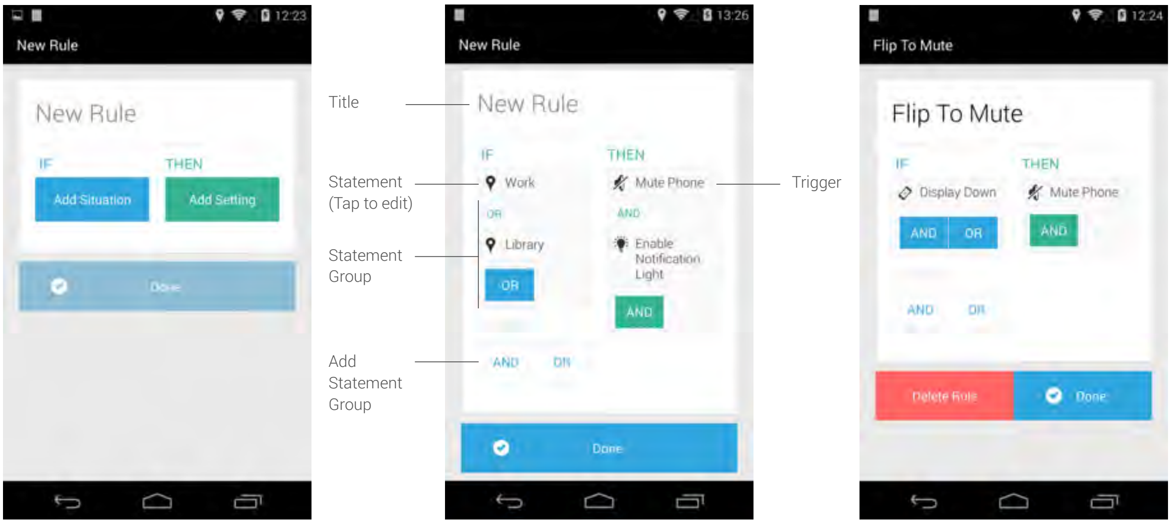


Figure 4.3.: Main user interfaces of the developed Android application.



(a) Initial state when creating a new rule.

(b) State after statements and triggers have been added.

(c) State when viewing or editing an existing rule.

Figure 4.4.: Different states of the `RuleActivity`.

The `RuleActivity` class's functionality can be configured by developers. Supported context groups and triggers can be added as follows:

```
// Add all supported Statement Fragments here
statementFragments.add(new LocationStatementFragment());
...
// Add all supported Trigger Fragments here
triggerFragments.add(new SoundTriggerFragment());
...
```

Tapping the "Add Situation", "Add Setting", "AND" or "OR") buttons, will show the `StatementsFragment` and `TriggersFragment` (described next), which will automatically show all supported context groups and triggers as selectable list items.

StatementsFragments & TriggersFragment (Selection)

The selection of statement context factors and modality triggers is realized as Android `DialogFragments`, which appear on top of the underlying `RuleActivity`. As described in Section 3.3.6 this is advantageous from a user perspective, as it allows to present the user an own view for each task without leaving the current context of the underlying `RuleActivity`. From an implementation perspective Android `Fragments` can be seen as modular sections of an activity having an own life cycle. They can be dynamically added or removed from the parent activity and receive own input events [1]. In the application this architectural concept allowed us to isolate the code for a multitude of different context factors and triggers in smaller maintainable fragments, which can still communicate with the parent `RuleActivity`.

Both, the `StatementsFragment` and the `TriggersFragment` provide a listener interface (`OnFragmentSelectedListener`), which is implemented by the parent `RuleActivity`. When the user selects a statement context factor or modality trigger, the `RuleActivity` is notified about this selection and presents the consequent `Statement/TriggerFragment`, e.g. the `LocationStatementFragment` (described in the following), to the user. This relationship is illustrated in Fig. 4.3.

StatementFragment

`StatementFragment` is a superclass which can be extended by the more specific context statement fragments (`LocationStatementFragment`, `LightStatementFragment`, `NoiseStatementFragment` etc.). It defines basic behavior which is common to all subclasses to avoid replicated code and provides a listener interface (`OnStatementChangedListener`) with the following methods:

```
public interface OnStatementChangedListener {
  // Called by RuleActivity when a statement was changed
  public void onStatementAdded(Statement statement);
  public void onStatementReplaced(Statement statement);
  public void onStatementRemoved(Statement statement);
}
```

The different context statement fragments implement this interface with their specific context factor and input parameters and pass the generated statement to the parent `RuleActivity`. The following example of the `LocationStatementFragment` illustrates this:

```
// Define context factor and group
LatLngContextFactor latLngContextFactor =
  new LatLngContextFactor(
    IContextGroup.CONTEXT_LOCATION,
    LocationContext.LATLNG_GET_LOCATION);

// Get input parameter
```

[1] http://developer.android.com/guide/components/fragments.html

```
LatLng coordinates =
  new LatLng(marker.latitude, marker.longitude);

// Create statement
Statement newStatement =
  new Statement(latLngContextFactor, LatLngOperator.equals(coordinates));

// Pass new statement to callback activity
callbackActivity.onStatementAdded(newStatement);
```

The following dialog fragments were implemented to allow users the definition of various contextual statements: `BatteryStatementFragment`, `LightStatementFragment`, `LocationStatementFragment`, `NoiseStatementFragment`, `OrientationStatementFragment`, `TimeStatementFragment`.

TriggerFragment

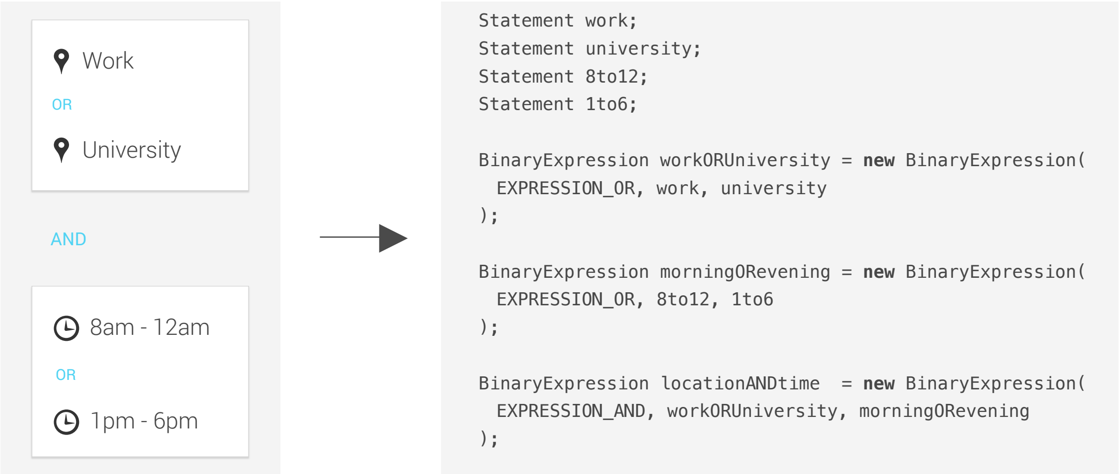
Similarly, the `TriggerFragment` superclass can be extended by the specific trigger fragments (`SoundTriggerFragment`, `VibrationTriggerFragment`, `BrightnessTriggerFragment` etc.). The follwing examples demonstrates the `SoundTriggerFragment`'s implementation of the callback interface:

```
// Define trigger
AudioTrigger audioTrigger = new AudioTrigger();
int action = ringerRadioGroup.indexOfChild(selectedRadioButton);
audioTrigger.setAction(action);

// Pass new trigger to callback activity
callbackActivity.onTriggerAdded(audioTrigger);
```

Translating Graphical Representations to Logical Expressions

A central part of the implementation is an algorithm which builds logical expressions (that can be evaluated by the evaluator) from their graphical representations in the GUI. Fig. 4.5 illustrates this with an example: the graphical representation consisting of two groups, each again consisting of two statements (left), has to be translated into an equivalent logical expression (right). The developed algorithm, which recursively connects statements and groups to a single logical expression is shown in the following pseudo code:



```
Statement work;
Statement university;
Statement 8to12;
Statement 1to6;

BinaryExpression workORUniversity = new BinaryExpression(
  EXPRESSION_OR, work, university
);

BinaryExpression morningORevening = new BinaryExpression(
  EXPRESSION_OR, 8to12, 1to6
);

BinaryExpression locationANDtime  = new BinaryExpression(
  EXPRESSION_AND, workORUniversity, morningORevening
);
```

Figure 4.5.: Graphical representation (left) with the equivalent logical expression (right).

```
ArrayList<LogicalExpression> groupLogicalExpressions;
LogicalExpression finalLogicalExpression;

// 1. Transform each group into a single logical expression
// and collect them in the groupLogicalExpressions list
FOR EACH group in statementGroups
  IF (group.size == 1)
    groupLogicalExpressions.add(statement)
  ELSE
    FOR EACH statement in group {
      binaryExpression = new BinaryExpression(
        group.getExpressionType(),
        binaryExpression,
        group.get(statement);
    groupLogicalExpressions.add(binaryExpression);

// 2. Create final logical expression from groups
IF (groupLogicalExpressions.size == 1)
  finalLogicalExpression = groupLogicalExpressions.get(0);
ELSE
  FOR EACH logicalExpression in groupLogicalExpressions
    finalLogicalExpression = new BinaryExpression(
      rule.getExpressionType(),
      finalLogicalExpression,
      groupLogicalExpressions.get(logicalExpression));
```

GUI Helper

A GUIHelper class allows to define human understandable descriptions for statements and triggers. A TimeContext statement should for example not be displayed as “time $>$ 12 and time $<$ 13”, but rather in a commonly understandable way such as “Time is between 12:00 and 13:00”. The helper class also allows to map particular parameter values to reasonable ranges. The system screen brightness for example takes integer values from 0 to 255. The GUIHelper maps these values to percentage values (0-100%) to make them understandable for the user.

4.2.2. Background Services & Evaluation Scheduling

Steadily observing context changes and adapting modalities without direct user interactions requires that the application is running in the background. This was realized through long-lasting background processes called Android Services, Android BroadcastReceivers which start these services when particular events (called Android Broadcast Actions) happen and the Android AlarmManager which schedules evaluation operations of the evaluator outside the lifetime of the application.

Initialization

To prevent the user from having to start the application once every time the mobile device is restarted, the InitializationReceiver waits for the BOOT_COMPLETED broadcast action (evoked by the Android system). Once the broadcast action has been received, the application will load (deserialize) all rules, initialize all context groups, and then send a RULES_INITIALIZED broadcast action as a last point. The SchedulingReceiver is registered to this action (as well as to the RULES_UPDATED action) to schedule the evaluation of rules once the initialization process has finished (see Fig. 4.6).

Scheduling

The Evaluator evaluates rules in a defined time interval. While the initial version of the *M3I* framework used a fixed update interval (of for example 1000 milliseconds), the developed application dynamically adapts the evaluation schedule based on the currently used context factors to reduce processing costs and energy impact. While, for example, the OrientationContext requires relatively short update intervals of maximum a few seconds to quickly react to orientation changes, update intervals in the range of minutes are sufficient for the TimeContext or LocationContext. This is made possible by allowing developers to define reasonable time intervals for each context group (e.g. 3 minutes for the LocationContext) which will then be used to calculate the minimum update interval for the evaluation schedule. Considering these differences can result in a highly increased update interval especially when only "slow" context groups are currently in use. Also event listeners (used for example for the light sensor and location manager) are dynamically registered and unregistered based on context groups which are actually used.

This functionality was implemented in the following way: whenever the set of rules is updated (meaning that rules have been added, deleted or modified) a `RULES_UPDATED` action is broadcasted. Once this broadcast is received by the `SchedulingReceiver` it will recalculate the minimum update interval in the described way, (un-)register (un-)needed event listeners and reconfigure an Android `AlarmManager`. The `AlarmManager` triggers the `EvaluationService`, which evokes the `evaluateOnce()` method of the `Evaluator` in the calculated update interval (see Fig. 4.6). Using the `AlarmManager` allows to make use of Android's *inexact repeating* mechanism which optimizes energy consumption by synchronizing repeating alarms from multiple applications and firing them at the same time, thus reducing the total number of times the system must wake the mobile device [1].

4.2.3. Laboratory and Field Study Versions

In order to evaluate different aspects of rule-based multimodal interaction two separate versions of the Android application with different feature sets were developed.

Laboratory Study Version

The application version prepared for the conduction of the laboratory study was extended with implementations of the three conceived user interface variants for the creation of rules, namely the *Balanced*, *Situation-oriented* and *Modality-oriented* rule layout (see Fig. 3.8) alongside with the possibility to execute time measurements for these competing layouts. Also GUI implementations of the two rule creation alternatives, namely the *Suggestion* and *Snapshot* were added. To simulate the machine learning algorithms which would have been required for a full implementation of the *Suggestion* method, we manually added two rules which appear appropriate for most persons to the GUI (see Fig. 3.12).

To simulate notifications about modality changes and the recognition of different input gestures in Wizard of Oz experiments an additional web application (using the Ruby on Rails framework[2] and Google Cloud Messaging (GCM)[3]) was developed. Together with an implemenation of the GCM service in the Android application the web application allowed us to remotely trigger *Notifications*,

[1] http://developer.android.com/training/scheduling/alarms.html

[2] http://rubyonrails.org

[3] http://developer.android.com/google/gcm/

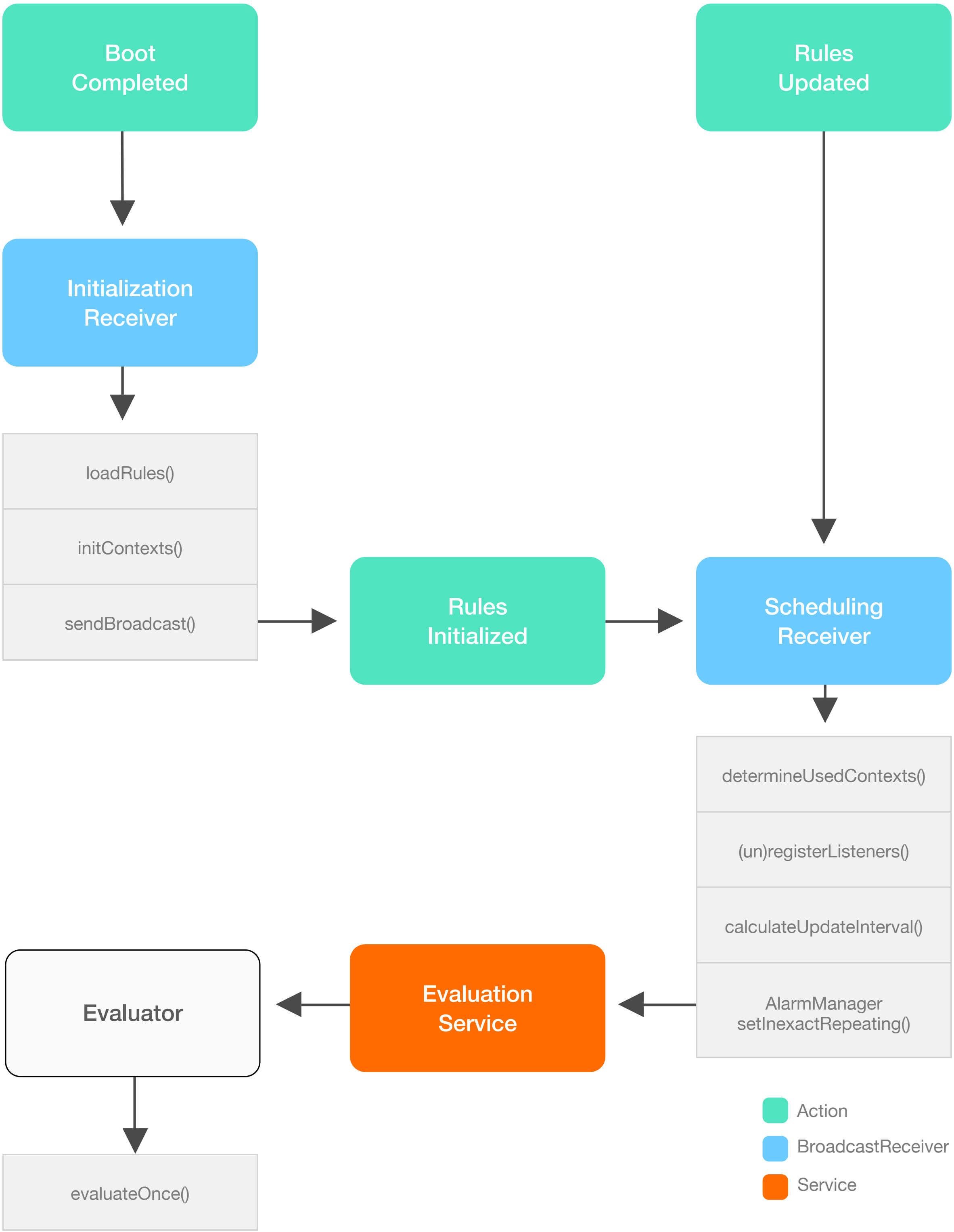


Figure 4.6.: `BroadcastReceivers` and `Services` running in the background of the application to ensure and optimize the continuous evaluation of rules.

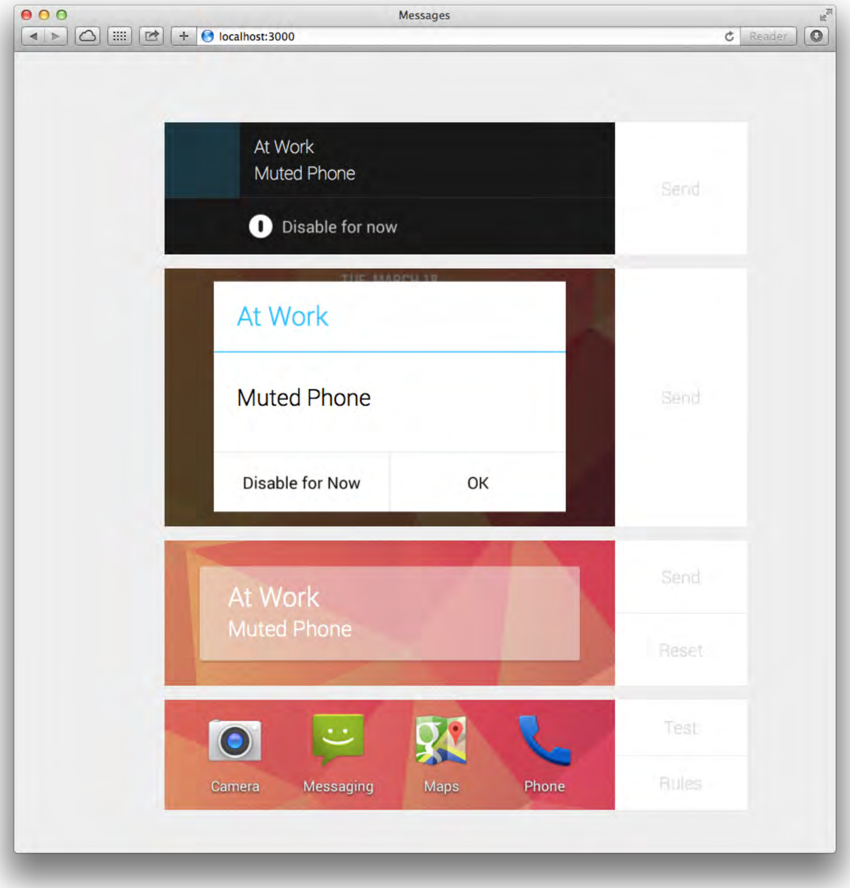


Figure 4.7.: Web application developed for the conduction of the laboratory study to remotely trigger *Notifications*, *Alerts*, *Widget* updates and applications (camera, messaging, maps and phone) on the test device.

Alerts, *Widget* updates and applications (camera, messaging, maps and phone) on the test device (see Fig. 4.7).

Field Study Version

The version used for the field study was a fully functional Android application featuring the *balanced* rule layout, which was tested over a period of approximately two weeks before handing it out to the participants. It was equipped with a self-written logging component to gather usage statistics (e.g. the number of created, deleted and disabled rules, and the selected context factors and modalities), which could be transmitted to a corresponding web server (also running on the Ruby on Rails framework). The implementation of the field study version applied modality changes in the background without notifying the user.

Chapter 5.

Evaluation

To better understand multimodal interaction and to evaluate the proposed rule-based approach a laboratory study and a field study were conducted. The general focus of the laboratory was to evaluate different user interfaces and concepts for the creation of rules and for providing awareness about modality changes. The first part consisted of questions which aimed to gain insights about the participants general usage of modalities: which modalities are frequently used, how do users switch between them, and how often do they apply modality changes? We also wanted to find out how well common ways to adjust modalities are suited to find generally appropriate settings and in which situations users commonly experience inappropriate modalities. To review the rule-based approach with regard to acceptance the participants were also asked about their opinion on the smartphone gathering context information in the background and about automatic adjustments of the smartphone's modalities. Another aim was to identify rules which appear particularly important to the participants in an exploratory interview. The research question in the second part of the laboratory study was to find the best method for creating multimodal interaction rules with relation to efficiency (time), effectiveness (errors), and satisfaction (ease of use, clarity, usability) with the help of a comparative experiment. Also the two previously described alternative rule creation methods were evaluated with regard to the participants' attitude, comfort and satisfaction. The research question in the next part was to find out which method to provide awareness about applied modality changes is preferred in terms of efficiency, effectiveness and satisfaction. To that end another experiment presenting different notification methods was developed. While these parts of the laboratory story were focused on output modalities the main research question of the last part was to evaluate participants' opinion on the usage of different input modalities. A Wizard of Oz experiment which allowed the participants to try out the conceptualized input methods by themselves was used for this. In this context another aim was the exploration of new ideas for rule-based input methods.

The general aim of the field study was to evaluate the long term usage of the proposed rule-based system and to gain insights about the contexts and modalities used most frequently. One research question was how many rules users create (and disable or delete) to describe realistic usage situations and how simple or complex the created rules typically are with regard to their length and structure. Also a daily questionnaire was created to collect feedback about the participants satisfaction with the application and its reliability on a daily basis.

5.1. Laboratory Study

5.1.1. Experiment Setup

The laboratory study was conducted during a two week period. The tasks were performed using an Android application featuring the different user interfaces to be evaluated as well as time measuring and small in-app questionnaires. The device used was an LG Nexus 4 running Android 4.4.2. Additionally the described web application was used by the instructor to trigger particular actions for Wizard of Oz experiments. A camera was used to audio-record the participants answers to oral questions and to video-record their interaction with the smartphone. Alongside the Nexus smartphone and the instructor's computer another laptop was used to present an online questionnaire for each study section to the participants. A remote statistics server was used to record the measured results after each task.

5.1.2. Participants & Demographic Data

For the laboratory study 24 participants were recruited. Six of them were female, the average age was 24.3 years (standard deviation: 2.5 years, range: 20-31 years, see Fig. 5.1). Most of the participants were students (83%), two participants were Designers and two other participants were from the medical sector. 42% reported the Abitur certificate as their highest educational achievement, 38% had a bachelor's degree, 17% a master's degree, one participant had a PhD degree (see Fig. 5.1). They were recruited using mailing lists and social networks and could choose between five Euros or one university credit point as reward. All participants had a western background.

The majority of the participants owned an Apple iPhone (58 %), 21% owned a smartphone by Samsung, the remaining hardware manufacturers were Nokia, LG and Sony (see Fig. 5.2). The participants were asked to estimate their technical skill with relation to smartphones on a five-point Likert scale (ranging from “No idea of technology”(1) to “Expert”(5)). One third of the participants rated their expertise with 3, another third with 4 and the last third with 5 so the average estimated skill was 4 (see Fig. 5.2).

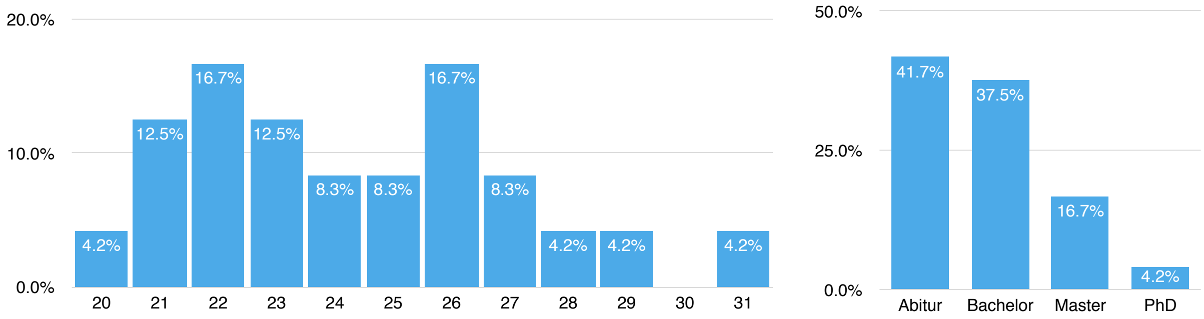


Figure 5.1.: Distribution of participants' age and educational degrees.

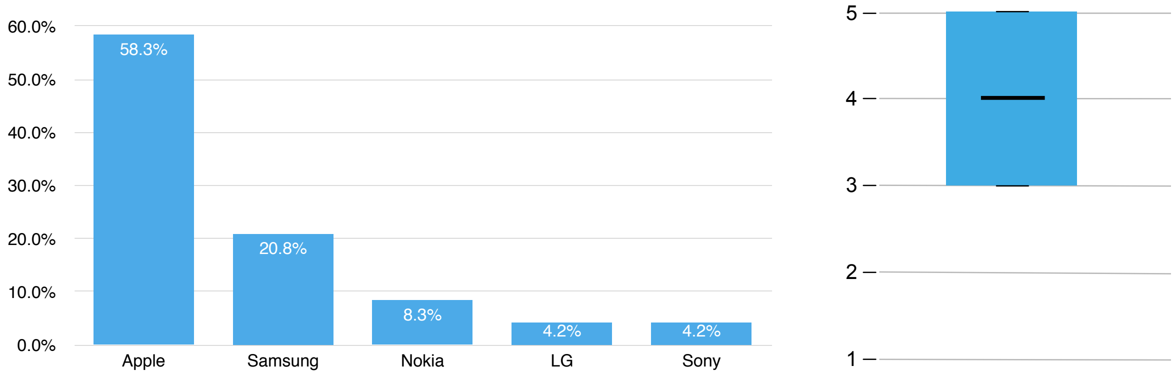


Figure 5.2.: Currently owned smartphone by manufacturer (left) and self-estimated technical expertise (right, the bold black line in the right diagram shows the median.

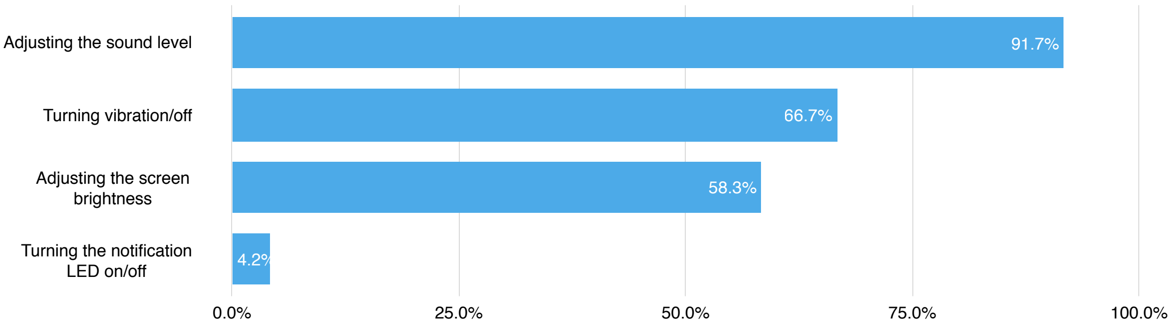


Figure 5.3.: Modality changes made on a regular basis.

5.1.3. General Questions about Modality Usage

After explaining the terms *context* and *modality* to the participants they were asked some introductory questions to gain insights about their general usage of modalities. The first part of the questionnaire addressed questions about *which* output modalities were regularly used, *how* users switch between different output modalities and *how often* they apply such changes. When being asked which possibilities of their smartphone they regularly used 92% of all participants reported that they regularly changed the sound level (see Fig. 5.3). This seems comprehensible as sound is amongst the most obtrusive output modalities (everyone knows how disturbing a ringing phone in a silent environment can be). Also it is generally very easy to adjust this modality as most smartphones have their own specific button to mute the phone. The majority of all participants also regularly controlled their smartphones vibration and screen brightness. Only 4.2% reported to regularly turn their smartphone's notification light on or off. It was striking that most participant's smartphone featured a notification LED (primarily used as a camera flash), but only a few knew about the ability to use it for visual notifications. Fig. 5.3 gives on overview of modality changes made on a regular basis by the participants.

The next question was, how often the participants change their smartphone's in-/output modalities on an average day. The results show that this is a task which is performed several times a day by the vast majority of all users: 42% stated that they make changes to their phone's modalities once or twice a day, one third reported to change modalities at least twice and 21% reported to change them at least five times a day. One participant reported that he never changed his smartphone's modalities (see Fig. 5.4).

Participants were furthermore asked how they change their smartphone's modalities. The results show that many different ways are used: 79% reported to use (hardware) buttons, 63% use their operating systems settings, the third mostly used possibility were widgets and control centers (29%). 21% reported to use apps to change modalities (see Fig. 5.5). One participant stated to use NFC tags placed in different locations in his apartment to trigger modality changes.

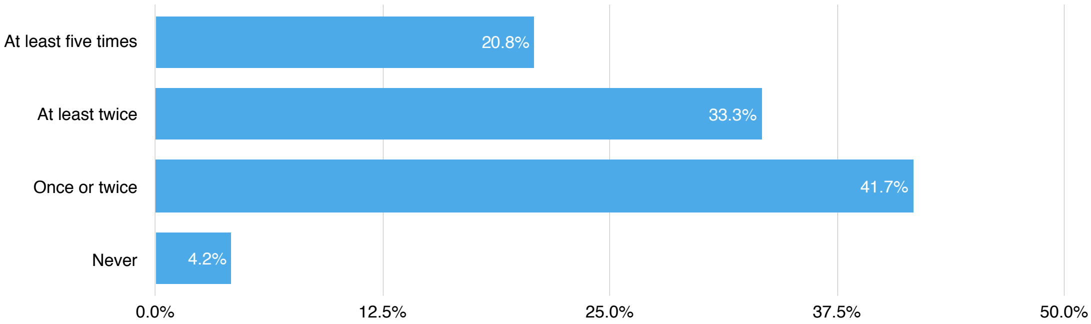


Figure 5.4.: Frequency of modality changes on an average day.

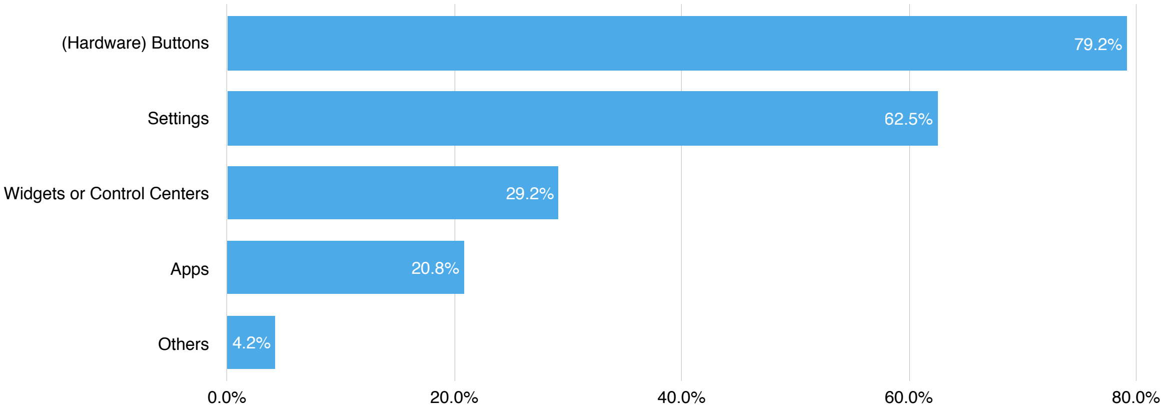


Figure 5.5.: Ways to change modalities.

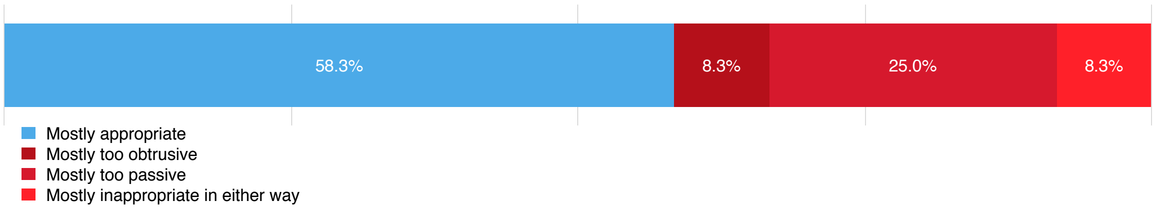


Figure 5.6.: General mode with regard to in/output modalities.

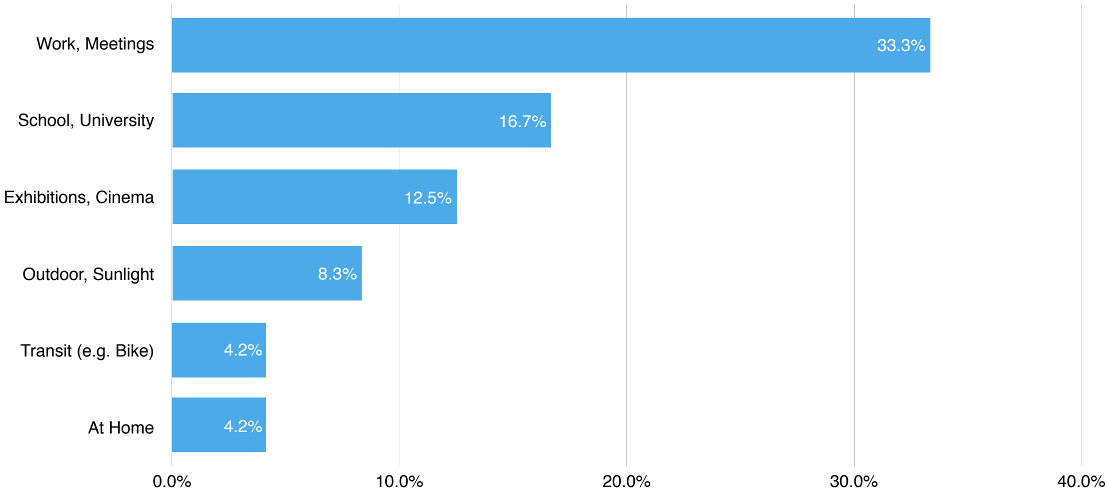


Figure 5.7.: Situations in which participants experience inappropriate modalities.

The next questions aimed to find out how well these common ways to adjust modalities are suited to find generally appropriate settings. While most participants reported their smartphone's general mode with regard to in/output modalities as mostly appropriate, almost half of the participants (42%) reported that their smartphone was mostly too passive (25%), too obtrusive (8%) or inappropriate in both ways (8%) (see Fig. 5.6).

Which are common situations in which users experience inappropriate modalities? The most frequently mentioned situations were *At work* and *In a meeting* (mentioned by one third of all participants), followed by the situations *At school* or *At university* (mentioned by 17%). 8% of all participants reported that they had experienced unfavorable modalities in *outdoor situations* (being unable to read the screen in strong sunlight). Other mentioned situations include *In transit* (e.g. while biking) and *At home* (the participant often forgot to return from a passive work setting when coming home in the evening). A summary is shown in Fig. 5.7.

Two important requirements for the proposed rule-based approach are that the smartphone continually gathers context information though sensors and that it is able to automatically adjust modalities in the background. To evaluate the acceptance of this behavior the participants were asked two questions: *What is your opinion on the smartphone gathering information about your*

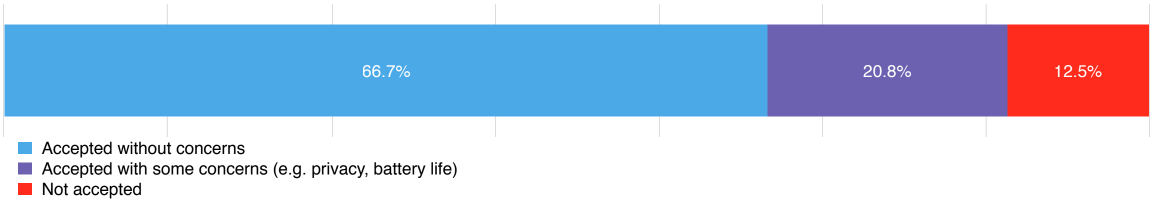


Figure 5.8.: Acceptance of the smartphone gathering context information via sensors in the background.

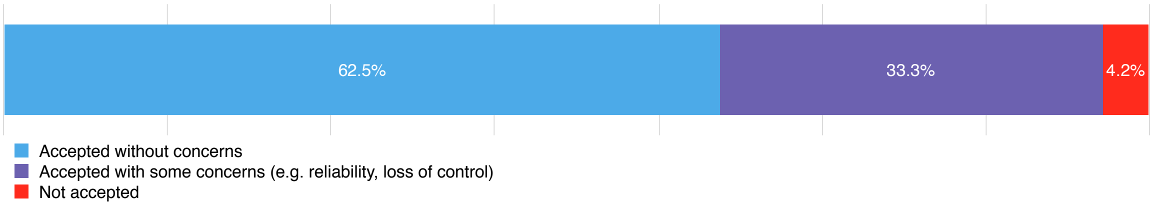


Figure 5.9.: Acceptance of the smartphone automatically adjusting modalities.

current context via sensors in the background? and *What is your opinion about the smartphone automatically adjusting modalities?*. 68% stated that they had no concerns if their smartphone gathered context information in the background. Different explanations were added: two participants stated that it was fine for them for this purpose, another two participants said that the smartphone did this anyways, another participant explained that the gathered context information was uncritical to him. 21% reported that the behavior was generally acceptable, but mentioned some concerns which would be important to them: two participants mentioned that the gathered information should not leave the device (and only be used locally), one participant stated that location data might be problematic for him in some cases, another participant said that his phone's battery life should not be effected heavily. 13% reported that it would not be acceptable at all for them if their smartphone gathered context information in the background (see Fig. 5.8).

Only one participant stated that he would not like his smartphone to automatically adjust modalities (see Fig. 5.9), while 63% had no concerns at all about such behavior and 33% reported that it would be generally acceptable for them under certain requirements. Three participants mentioned high reliability as a requirement, four participants said that the ability to control modalities on their own would be important to them.

To gain more specific insights on the participants opinion they were furthermore asked about their opinion on different concrete scenarios (see Fig. 5.10). Most presented rules were accepted by more than 75% of all participants. The scenario *My smartphone switches to the notification light instead of ringing when I put it on the desk with the display facing down* received neutral responses by 38%, which corresponds to the results of 5.3 showing that the notification light is simply not used by the majority of all participants. The highest percentage of rejections was given to the rule

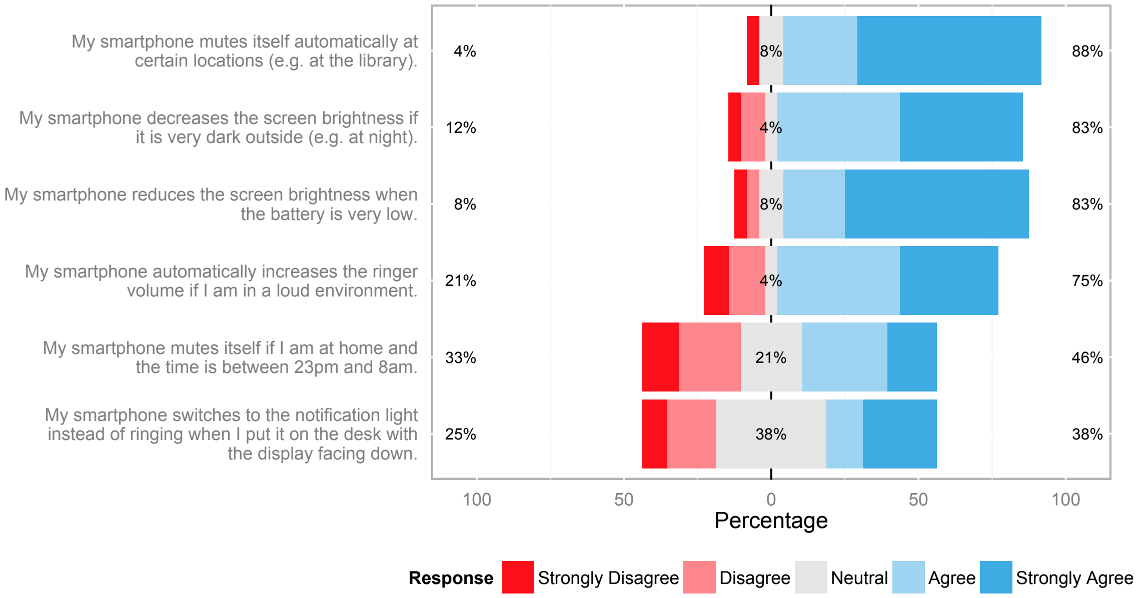


Figure 5.10.: Acceptance of different rule scenarios.

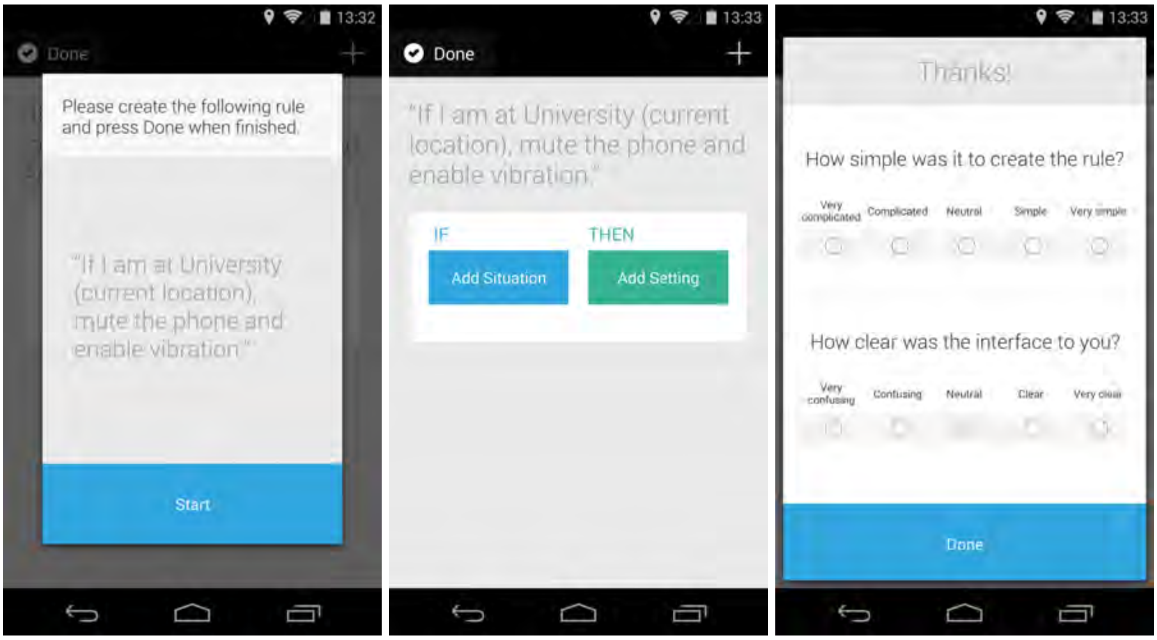
My smartphone mutes itself if I am at home and the time is between 23pm and 8am. A possible reason for this might be that this rule is relatively restrictive (due to the precise time statement) and might therefore not match the usage behavior of many users.

5.1.4. Rule Creation

As previously described we designed a *Balanced*, *Modality-oriented* and *Situation-oriented* user interface (see Fig. 3.8) for the creation of multimodal interaction rules. The research question in this section of the study was to find the best layout method with regards to efficiency (time), effectiveness (errors), and satisfaction (ease of use, clarity, usability).

Procedure

This part of the laboratory lasted approximately fifteen minutes. Each participant was asked to perform six tasks: for each user interface variant (Fig. 3.8) two different rules should be created. At the beginning of each task the participants were shown a short introduction reading the rule to be created, e.g. "If I am at university (current location), mute the phone" (Fig. 5.11a). After pressing the start button a time measurement was started in the background. The rule description was also permanently shown at the top of the view during the task to avoid unwanted side effects by persons suddenly forgetting parts of the rule to be created (Fig. 5.11b). After each task the participants had to rate the current user interface concerning ease of use and clarity on a five-point Likert scale to gather immediate feedback. Additionally a System Usability Scale (SUS)



Please create the following rule and press Done when finished.

"If I am at University (current location), mute the phone and enable vibration"

"If I am at University (current location), mute the phone and enable vibration"

Thanks!

How simple was it to create the rule?

How clear was the interface to you?

(a) Step 1: Introduction

(b) Step 2: Task

(c) Step 3: Questions

Figure 5.11.: Steps of each rule creation task, exemplary shown for a task with layout *Balanced* and rule type *1*.

questionnaire had to filled out after every second task (before continuing with the next variant).

Design

A repeated measures within participants design was used. The first independent variable was the *layout variant* with three levels (*Balanced*, *Modality-oriented*, *Situation-oriented*). There are different types of rules with regard to their structure, which influences how easy they are to create using the different variants. The *Modality-oriented* variant has a slight advantage for rules where different situations are assigned to one modality (structure: *modality X* ↔ *situation A and/or B*) as the layout requires less taps for this. The *Situation-oriented* variant has a slight advantage for rules where several modalities are assigned to one situation (structure: *situation A* ↔ *modality X and Y*) as the layout requires less taps for this. Table 5.1 summarizes these differences. To account for this factor *rule type* was added as a second independent variable (the two levels being *Rule 1* and *Rule 2*). *Layout variant* was counterbalanced based on a Latin Square Design, *rule type* was alternated. The dependent variables were *efficiency* (execution time in ms), *effectiveness* (error and success rate) and *satisfaction* (ease of use, clarity, usability (SUS)). The null hypothesis was that all three layout variants are equally good concerning efficiency, effectiveness and satisfaction.

Results: Efficiency

As described above the time to create rules was measured to determine the most efficient layout. Creating both rule types for the three different layouts resulted in six time measurement for each participant. With two-way ANOVA, we found significant effects of layout type $(F(2, 138) =$

Table 5.1.: Rule types chosen for the rule creation task of the laboratory study. The underlined entries indicate the layout which shows an advantage when creating the given rule type.

	Rule Type 1	**Rule Type 2**
Structure	Situation A ↔ Modality X and Y	Modality X ↔ Situation A or B
Example	*If I am at University, mute the phone and enable vibration*	*Set sound to ringing, if I am on the bike or it is very loud*
Advantage (according to minimum number of taps required to create the rule type)	*Situation-oriented* layout: 6 *Modality-oriented* layout: 8 *Balanced* layout: 10	*Situation-oriented* layout: 8 *Modality-oriented* layout: 7 *Balanced* layout: 10

$6.41,\ p < 0.01,\ partial\ \eta^2 = 0.017)$ and rule type $(F(1, 138) = 4.38,\ p < 0.05,\ partial\ \eta^2 = 0.592)$ on the execution time. As apparent from diagram 5.12 the *Situation-oriented* layout did best here with a mean of 17.9 seconds for *Rule 1* and a mean of 26.5 seconds for *Rule 2*. The *Balanced* and *Modality-oriented* layout were almost equally fast for *Rule 1* (means: 28.7 and 29.8 seconds), but the latter showed better results for *Rule 2* (means: 34.8 versus 29.3 seconds). To determine between which layout types significant differences exist a Tukey's Post-hoc test was performed. The test showed significant differences between the *Situation-oriented* and *Balanced* layout $(p < 0.01)$ and between the *Situation-oriented* and *Modality-oriented* layout $(p < 0.05)$. No significant differences were found between the *Balanced* and *Modality-oriented* layout. Taken together the *Situation-oriented* layout emerged as the most efficient layout, followed by the *Balanced* and *Modality-oriented* layout. When analyzing the video recordings of the participants creating the different rules the most apparent reason for the advance of the *Situation-oriented* layout was that users had to go through less dialogs when selecting a modality. The modality panel on the right of the screen (Fig. 3.8c) allows to directly select the desired modality without navigating through a series of different views.

Results: Effectiveness

The effectiveness of the different layouts was determined by the success rate (meaning how many rules were created correctly) and the error rate (meaning the average number of errors made for each layout). A rule creation task was counted as successful if the participant made no errors or corrected the error(s) she or he made. The most frequent error made was selecting and AND connection instead of an OR operator. Other errors made were selecting the wrong state for a modality (e.g. vibration ON instead of OFF), adding situations that were not required and selecting an incorrect value for the ambient noise threshold (the task required to define a loudness of more than 100 dB). All errors which were not corrected and resulted in invalid rules did not

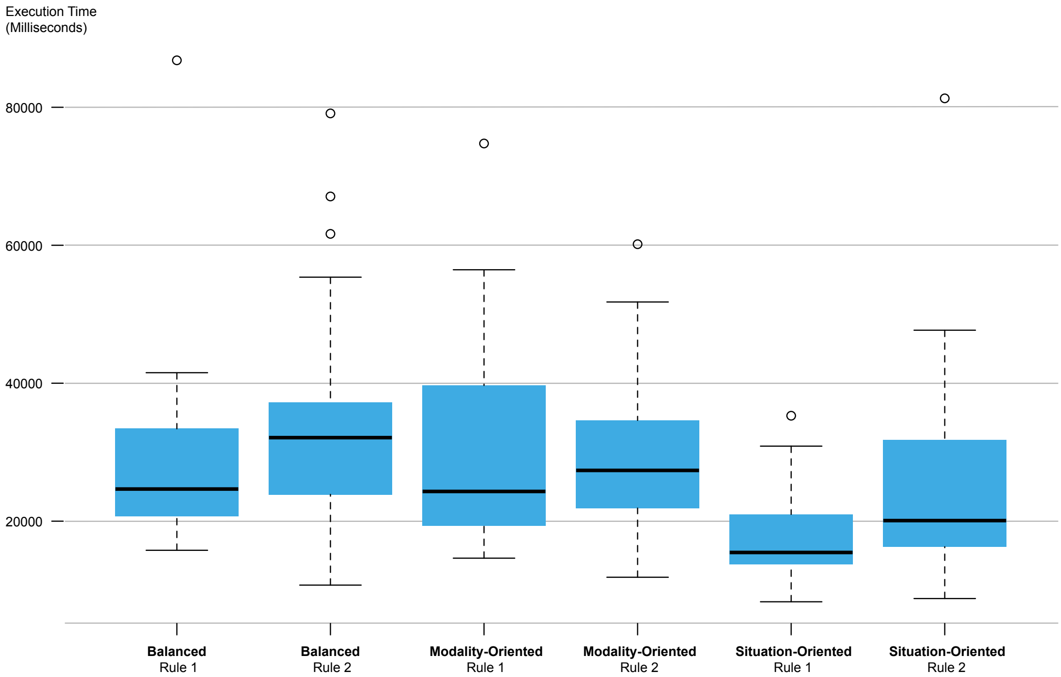


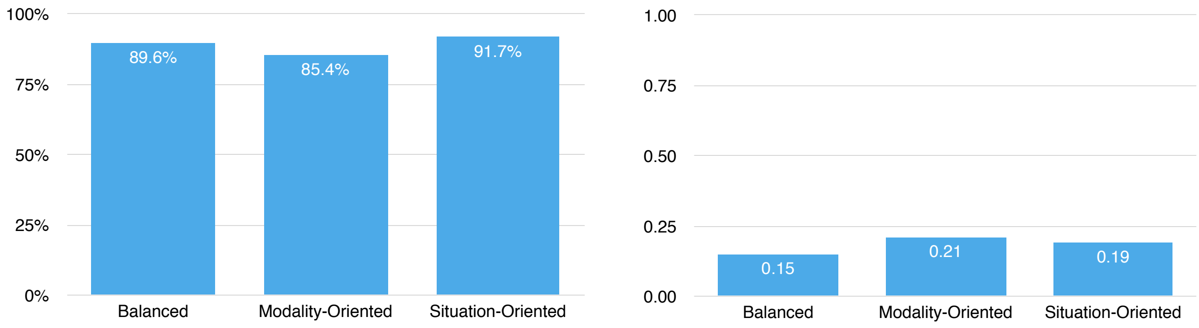
Figure 5.12.: Comparison of the three layout variants *Situation-oriented*, *Modality-oriented*, and *Balanced* by execution time. The bold black line in each bar shows the median.

lead to time advantages (as not adding required parts to a rule would), so that all results could be used for the efficiency analysis described above. The *Situation-oriented* layout showed the best success rate (91.7%) and the second best average number of errors (0.19). Vice versa the *Balanced* layout performed best with regards to errors (average number: 0.15) and second best with regards to the success rate (89.6%). The *Modality-oriented* layout showed the worst results with a success rate of 85.4% and 0.21 as the average number of errors (see Fig. 5.13 for all results). When analyzing the video recordings the most striking reason for this outcome might be found in the relatively high complexity of the *Modality-oriented* layout (showing one button for each modality) which could lead to confusion for some participants. This matter is also apparent in the the results of the SUS questionnaire where this layout was rated as the most complex one (see statement *I found the system unnecessarily complex* in Fig. 5.14).

A repeated measures ANOVA was performed to analyze the influence of the layout and rule type on the success and error rate. The results show that the differences between the layouts are not significant regarding both rates.

Results: Satisfaction

The participants' satisfaction with the different user interfaces was determined using the System Usability Scale (SUS), a simple ten-item scale which was created to evaluate the usability of digital systems [34]. With SUS scores of 86.7 (*Situation-oriented*), 78.4 (*Balanced*) and 74.9 (*Modality-oriented*) the last two layouts were rated as good (between 71.4 and 85.5) while the



(a) Success rate for each layout (meaning how many rules were created correctly).

(b) Average number of errors made for each of the three layouts.

Figure 5.13.: Comparison of the three layout variants *Balanced*, *Modality-oriented* and *Situation-oriented* by success rate and average number of errors made by the participants.

Situation-oriented layout was rated as excellent (greater than 85.5) according to [35]. Diagram 5.14 shows a comparison of the three layouts regarding the ten items of the SUS. The *Situation-oriented* layout scored best for eight of the ten items while the *Modality-oriented* scored best for the two remaining items (*I needed to learn a lot of things before I could get going with this system* and *I think that I would need the support of a technical person to be able to use this system*). The diagram (Fig. 5.14) shows that the former layout has the greatest advance to the other two user interfaces concerning ease of use (*I thought the system was easy to use*) and turned out the least cumbersome (*I found the system very cumbersome to use*).

With one-way ANOVA, we found a significant effect of layout type $(F(2,717) = 14.79,\ p < 0.0001,\ partial\ \eta^2 = 0.040)$ on the SUS score. To determine between which layout types significant differences exist a Tukey's Post-hoc test was performed. The test showed significant differences between the Situation-oriented and Balanced layout (p $<$ 0.01) and between the Situation-oriented and Modality-oriented layout (also p $<$ 0.01). No significant differences were found between the Balanced and Modality-oriented layout.

As described above and shown in Fig. 5.11c the participants furthermore rated the current user interface concerning ease of use and clarity on a five-point Likert scale to gather immediate feedback after each rule type. Here the *Situation-oriented* layout also scored best with an average value of 4.4 for both measurements.

An ANOVA with the collected values for ease of use and clarity showed that the layout has a significant effect in both cases. With two-way ANOVA, we found a significant effect of layout type $(F(2,138) = 3.78,\ p < 0.05,\ partial\ \eta^2 = 0.051)$ on the value for ease of use. We also found a significant effect of layout type $(F(2,138) = 3.78,\ p < 0.05,\ partial\ \eta^2 = 0.051)$ on the value for clarity. No signifificant effects were found for the rule type.

The results of a Post-hoc Tukey's test showed that the *Situation-oriented* layout scored significantly better compared to the *Balanced* and *Modality-oriented* layout ($p < 0.1$), while no significant differences were found between the *Balanced* and *Modality-oriented* layout (also for both ease of use and clarity). Analyzing the video recordings of the participants these findings can be explained with the compact representation of the *Situation-oriented* layout in comparison to the the *Modality-oriented*, which makes it easy to access all controls in the upper half of the screen and the smaller amount of dialogs to pass in comparison to the *Balanced* layout, which can make the layout less cumbersome for the users.

Results: Summary

The evaluation showed that the *Situation-oriented* layout is the best with regard to efficiency and satisfaction. Concerning effectiveness this layout shows slightly more errors than the *Modality-oriented* layout, but still has a superior success rate. The differences between the layouts regarding effectiveness are however not significant. Taken together the *Situation-oriented* layout performed significantly better than the other layouts regarding two of the three metrics so that the null hypothesis can be rejected. All results are summarized in Table 5.2.

To compensate for advantages the different layouts have when creating particular rule structures (see Table 5.1) rule type was added as an additional independent variable. This has shown to be not strictly necessary as *Rule 1* achieved better results than *Rule 2* regarding all three metrics (no statement can be made about the SUS score as this questionnaire was summarized for both rule types). Still the introduction of different rule types has led to more interesting measurements and observations.

Table 5.2.: Summarized comparison of the three evaluated layouts. The best results for each category are highlighted in blue. Measurements with statistically significant results are marked with a *.

	Balanced	**Modality-oriented**	**Situation-oriented**
Efficiency Execution Time (Mean)*	 31.7s	 29.6s	 22.2s
Effectiveness Success Rate Number of errors (Avg.)	 89.6% 0.15	 85.4% 0.21	 91.7% 0.19
Satisfaction SUS Score* Ease of Use (Likert 1-5)* Clarity (Likert 1-5)*	 78.4 4.0 3.9	 74.9 4.0 4.1	 86.7 4.4 4.4

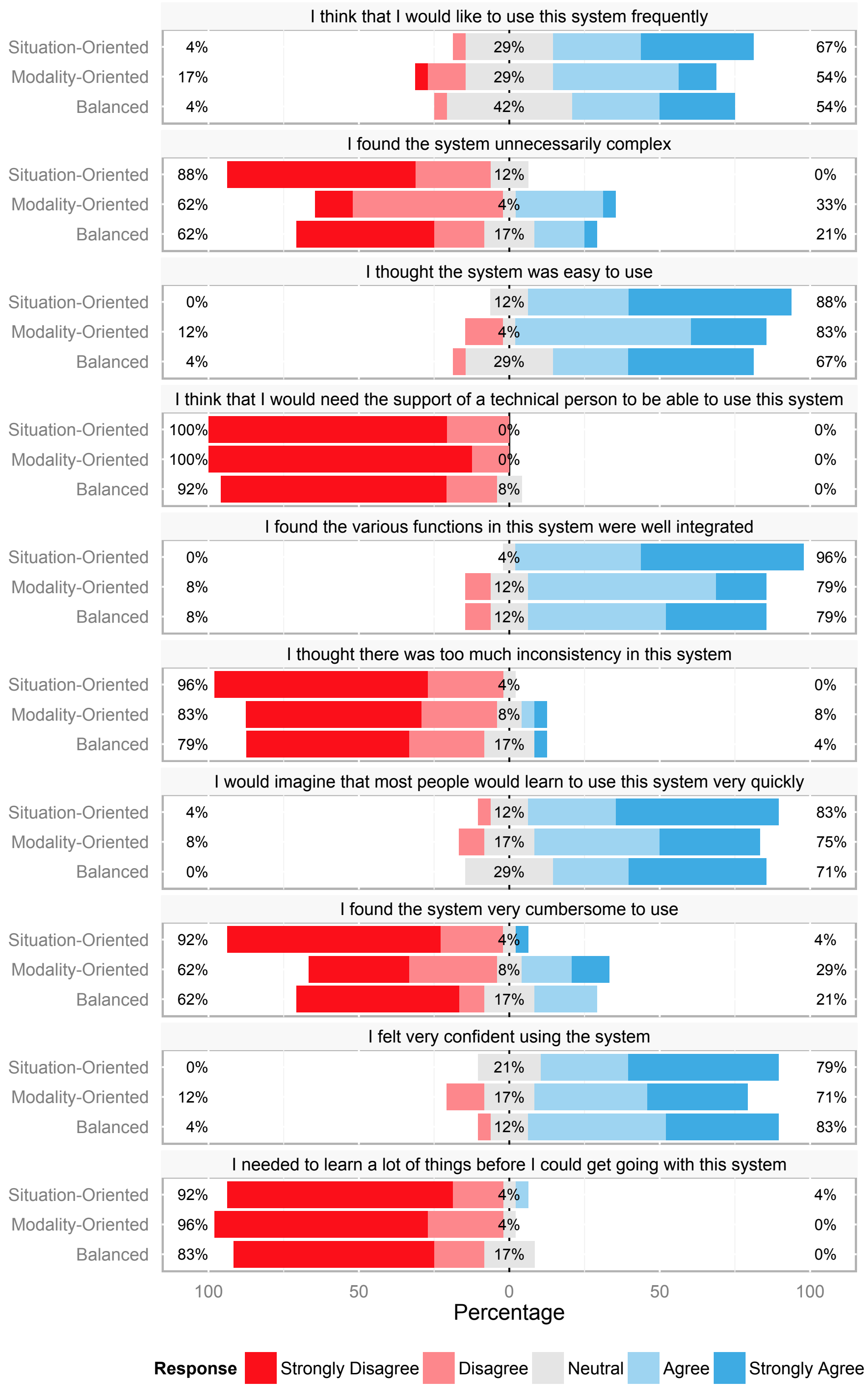


Figure 5.14.: Comparison of the three layout variants *Situation-oriented*, *Modality-oriented*, and *Balanced* by their results in the SUS questionnaire.

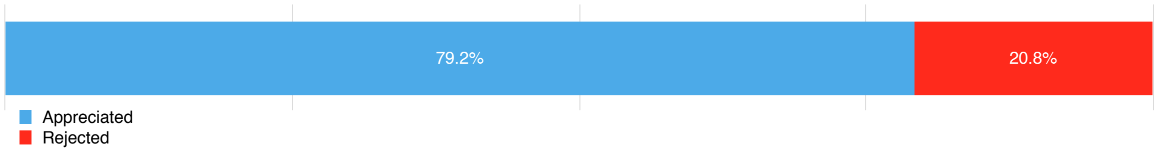


Figure 5.15.: Acceptance of the rule creation alternative *Suggestion*

5.1.5. Rule Creation Alternatives

In the next part of the laboratory study the two presented rule creation alternatives were evaluated. The idea of the alternative *Suggestion* is based on a machine learning approach: while a person is using her or his smartphone the app could register if certain modalities are frequently used in particular situations (see Fig. 3.12a). The second alternative (*Snapshot*) presents a pre-built rule containing a complete set of context factors and all currently selected modalities to the user (see Fig. 3.12b). The research question in this section of the study was to gain insights on the user's attitude, comfort and satisfaction for these methods. Participants could try both user interfaces and were then asked about their opinion in short interviews. The questions asked were: *What is your opinion on this approach?* and *Do you find this rule creation alternative helpful?*.

Results: Suggestion

This approach *Suggestion* was well accepted: 79.2 % of all participants appreciated this rule creation alternative (see Diagram 5.15). Participants liked that it could take work out their hand and that the approach might help them to find rules for recurring situations they were not actively aware of. It was also positively mentioned that the suggested rules could serve as an inspiration for new rules. One participant mentioned that suggestions appear as a good middle course between manual rule creation and a fully automated system (which would create and apply rules by itself) to him because it leaves control and an overview over active rules to the user. Participants who rejected the *Suggestion* based approach mentioned that they were skeptical it worked well. Others reported that they find this alternative superfluous and preferred to create rules manually. One participant stated that he would not want the app to record his usage behavior in the background.

Results: Snapshot

The participants had divided opinions on the rule creation alternative *Snapshot*: it was appreciated by 50% while the other half rejected it (see Diagram 5.16). Participants who liked the approach mentioned that it gave them a good overview over available possibilities. Some participants positively mentioned that they find it easier to remove (unwanted) parts of a new rule than to manually add them one after one. One participant stated he liked this alternative, because it does not require any machine learning but still behaved intelligently in determining the current situation and modalities. Participants who rejected the *Snapshot* approach reported that it presented too much information to them and that it seemed overly complicated to them. As in the *Suggestion*

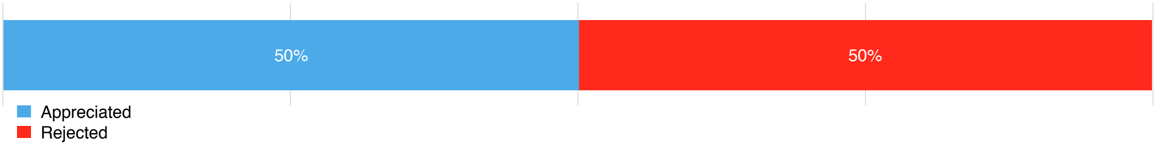


Figure 5.16.: Acceptance of the rule creation alternative *Snapshot*

based alternative some participants indicated that they preferred to create rules manually. It was furthermore negatively mentioned that the presented snapshot contained too many context factors which might not be used very frequently (e.g. ambient noise).

5.1.6. Interviews

After creating rules by themselves the participants were asked a series of general questions about their opinion on different areas of the rule creation process. To gain insights about rules which appear important to users the participants were first asked about a useful rule which comes to their mind spontaneously. The app dialogs showing the different available context and modality options were demonstrated beforehand so that the participants could recall all possibilities. 83% could spontaneously name a rule, more than 50% even came up with several rules. It was noticeably that many participants (50%) quickly thought about the situations "At Work" and "At University" for rules that would then apply passive modalities ("mute the phone", "only vibrate") (see topmost bar in Fig. 5.17). Contrary another frequently mentioned rule (21%) was to apply active modalities ("set to ringing", "enable vibration") when being at home. Several participants mentioned that they often forgot to revoke a passive setting when coming home and therefore missed calls or other notifications. Another common rule was to switch to active modalities when being on the bike or walking (mentioned also by 21%). Every eighth participant stated that she or he would like her or his smartphone to reduce the screen brightness when the battery is low. Several participants furthermore mentioned rules which would adapt their smartphone's modalities to loud environments and high ambient light levels (e.g. bright sun light) by adjusting the sound and brightness levels accordingly. One participant said that a rule which automatically mutes her phone at night would be useful her. Another participant proposed a rule which mutes the phone when being put on the desk with the display facing down ("flip to mute"). Taken together, one the one hand a relatively small set of frequently requested rules is recognizable, on the other hand all available context options (location, mode of transportation, time, ambient noise and light, battery level, orientation) were included.

To gain insights on the selection of context options offered by the app the participants were furthermore asked whether they could sufficiently describe relevant situations by using them. This was clearly affirmed by all participants. Another exploratory question was if there are any

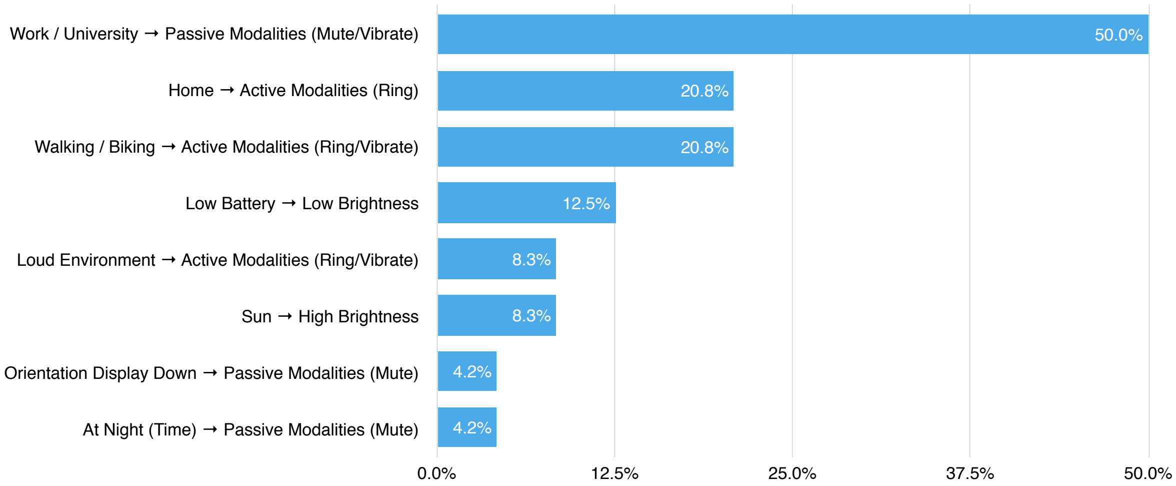


Figure 5.17.: Common useful rules as named by participants.

additional context factors which might be useful to the participants (independent from their technical feasibility). Different interesting ideas emerged:

- One idea was to summarize different places to **location categories** (e.g. cinema or restaurant) which could then be selected (instead of adding several locations connected over OR operators).
- One participant mentioned **weather & temperature** as an additional context option.
- Another thought was to determine whether the smartphone is currently located in the owner's **pocket** or whether it is hold in the **hand** at the moment.
- Nearby persons belonging to different **social circles** were mentioned by one participant as a useful addition to the available context options.
- Lastly **calendar events** were brought up as an additional possibility to describe a context apart from the current time. Particular appointments could then be associated with suitable modalities directly in the calendar.

Participants were furthermore asked whether they feel being able to express personally meaningful situations with the offered systems of AND and OR connections in a sufficient way. This was strongly confirmed by all participants. Only one participant with a less technical background mentioned that the difference between the two boolean operators was not always clear to him. This incidence will later be discussed in Section 5.3.

A rule clearly defines how modalities are changed when a user enters a defined context. There are, however, different possibilities to react when the user leaves a context again. One possibility would be to just keep the rule's modality settings applied even when it is not valid anymore, only a consequent rule would then change modalities the next time. A second possibility would

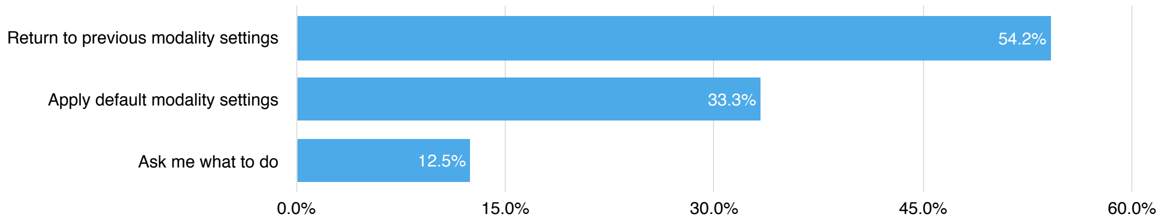


Figure 5.18.: Preferred ways of the smartphone to react when a user leaves a defined context.

be to apply a previously defined set of default modality settings (e.g. sound ringing, vibration on, screen brightness 100%, notification LED off). Another possibility would be to automatically return to the previously selected modality setting (e.g. if the smartphone is initially set to silent and the owner enters and then leaves a loud environment where the sound is set to ringing, the smartphone would automatically return to the previous silent setting). We asked the participants which modality setting they would expect after leaving a defined context. As shown in Fig. 5.18 the majority of all participants (54.2%) stated that they would expect the smartphone to automatically return to the previous modality setting, while 33.3% said that they would expect a defined default setting to be applied. 12.5% of all participants indicated that they would liked to be asked about a desired modality setting by the mobile device whenever they leave a context.

5.1.7. Awareness

As previously described, there are different ways to provide awareness on modality changes to the user: *Notifications*, *Alerts* or a *Widget*. Regarding the interaction with each of these awareness methods two possibilities emerge: a) once a rule becomes effective the system can either explicitly ask the user before changing modalities (*opt in*, e.g. Mute the phone?) or b) just apply the modality changes but offer the possibility to disable a modality for a given situation (*opt out*, e.g. Muted the phone. Disable for once?). An overview of all these different methods is given in Table 3.11. The research question in this part of the laboratory study was which method is preferred in terms of efficiency, effectiveness and satisfaction.

Procedure

Each participant was presented with the three notification types consecutively while performing a side task to simulate a more realistic usage situation. The side task was to find a particular photo in a photo gallery widget on the home screen. The side task was designed to be unsolvable (the photo to look out for did not exist) to prevent quick users from finishing too early. The actual task was to tap the disable button of each notification type (thus suppressing the modality change) as soon as the notification was noticed. Once a notification appeared on the screen a time measurement was started in the background. Tapping the disable button would then stop the timer.

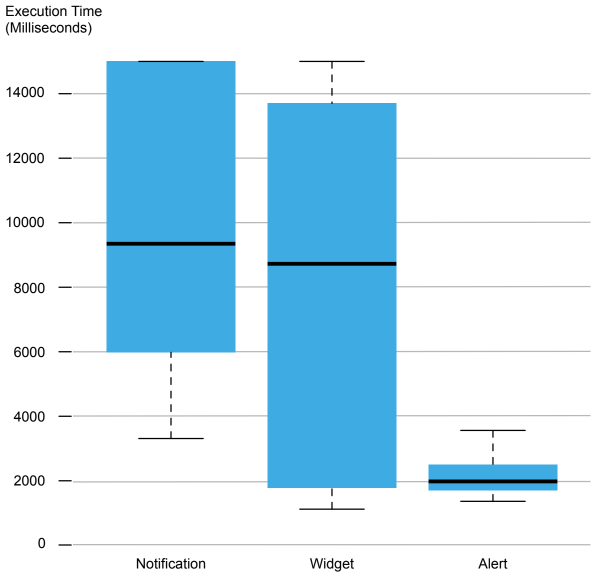


Figure 5.19.: Comparison of the three notification types Notification, Widget and Alert by execution time (ms). The bold black line in each bar shows the median.

Design

A repeated measures within participants design was used. The independent variable was the notification type with three levels (*Notification*, *Widget*, *Alert*). Notification type was counterbalanced resulting in six different order possibilities. The dependent variables were efficiency (execution time in ms) and effectiveness (error and success rate). The null hypothesis was that users can interact with displayed modality changes equally fast in all three notification types with equal success and error rates.

Results: Efficiency

The time to interact with a displayed modality change consists of two major components: firstly the user has to notice the change, secondly she or he has to interpret the displayed information and react on it properly.

With one-way ANOVA, we found a significant effect of notification type $(F(2,69) = 21.43,\ p < 0.0001,\ partial\ \eta^2 = 0.984)$ on the time to react on a modality changes. As visible in diagram 5.19 *Alert* was the most efficient method with a mean of 2.1 seconds, followed by *Widget* (mean: 7.6 seconds) and *Notification* (mean: 9.9 seconds). If participants did not notice one of the notification types the task was aborted after 15 seconds. In this case the result time for the task was set to that duration.

A Tukey's test was performed to determine between which methods significant differences exist. The test showed significant differences between *Notification* and *Alert*, and between *Widget* and *Alert* (in both cases $p < 0.0001$). There were no significant differences between *Widget* and *Notification*.

Results: Effectiveness

The effectiveness of the different notification types was determined by the success rate (meaning that a modality change was dismissed correctly) and the error rate (meaning the average number of errors made when interacting with each notification type). The *Alert* and *Widget* showed the best success rates (100% and 87.5%), while only 54.2% noticed when the modality change was shown to them as a *Notification* and dismissed it correctly using the disable button.

With one-way ANOVA, we found a significant effect of notification type $(F(2, 69) = 10.83,\ p < 0.0001,\ partial\ \eta^2 = 0.239)$ on the success rate. A Post-hoc Tukey's test showed significant differences between the combinations *Notification-Alert* and *Notification-Widget* $(p < 0.005$ in both cases). There was no significant difference regarding the the success rate between *Alert* and *Widget*. The participants made no errors when interacting with the *Widget*. The average number of errors made when interacting with the Notification was 0.08 (the participants first tried to tap the notification icon instead of pulling it down from the status bar), the average number of errors for the Alert was 0.04 (the participants tapped the "OK" button instead of "Dismiss"). A repeated measures ANOVA however showed that these findings are not significant.

Results: Satisfaction

As usabilities measurements like the SUS questionnaire, which was used to evaluate the rule creation user interfaces, are not well suited for the relatively short and simple interaction with the different notification types, the participants were interviewed about their satisfaction with the different notification methods. After a demonstration and explanation of the available possibilities (see Fig. 3.11) the participants were asked which method they preferred and why they do so. More than one third of all participants (37.5%) reported that they would prefer to get a *Notification* with the optional possibility to *opt out* of modality changes in unusual situations. 4.2% preferred this notification type in combination with the interaction type *opt in*. Participants mentioned that they liked notifications for their appropriate level of obtrusiveness: while always being visible in a subtle way they do not disturb other ongoing actions too much (see Fig. 5.20). 25% stated that they would prefer the *Widget* with the possibility to *opt out* as a way to become aware of modality changes. It was appreciated as a central point to look for currently applied rules without actively disturbing the user. Participants who did not like this notification type reported that it would take too much space on their smartphone's homescreen. A smaller part preferred the *Alert* as an appropriate to get notified (12.5% with the possibility to *opt out*, 8.3% with the necessity to *opt in* first). While the *Alert* notification type was liked for the high level of control and its noticeable appearance by some participants, the majority of all participants described this method as too obtrusive and distracting. Generally the majority of all participants argued in favor of the *opt out* interaction type while only 12.5% preferred to *opt in* to modality changes. The proponents of the former interaction type argued that not applying modality changes automatically would make the whole system less attractive to them, because it would require a similar amount of work as changing modalities manually. Participants also felt that by creating a rule they had already

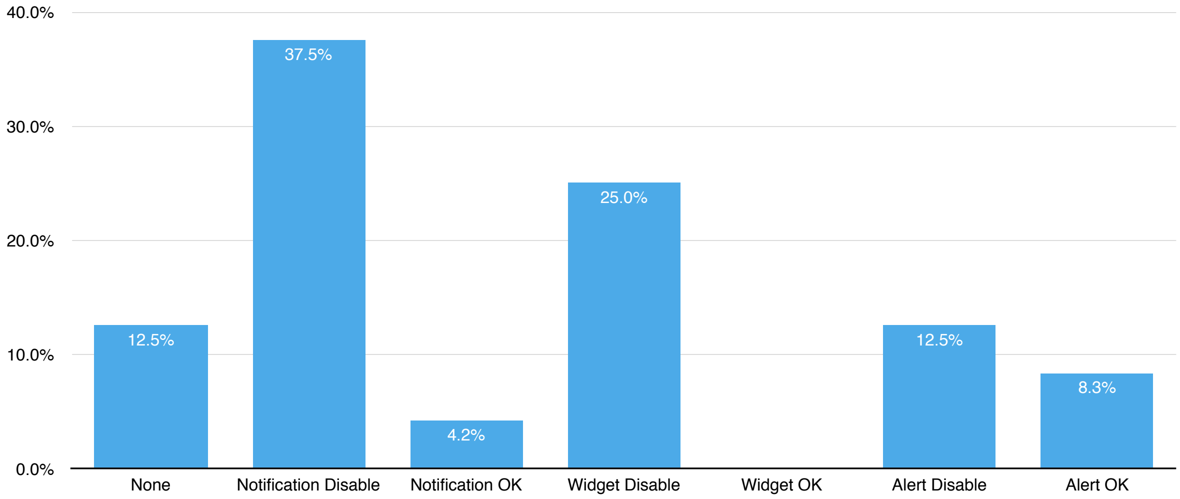


Figure 5.20.: Preferred ways to get informed about modality changes

approved of resulting modality changes so that no further agreement was necessary for them. 12.5% of all participants reported that they would not want any form of notifications for this reason.

Results: Summary

While the *Notification* awareness method did not score best with regard to efficiency and effectiveness it was preferred by more than 40% of all participants. The efficiency when interacting with the different notification types clearly correlates with its level of obtrusiveness which was however a negative feature for several participants. These findings suggest that non-obtrusiveness is important to the participants than efficiency. Taken together the results show that the preferred notification type is very subjective from person to person. The null hypothesis can be rejected as significant differences were observed. There is however no clearly superior method to provide awareness about modality changes. All results are summarized in Table 5.3.

5.1.8. Rules for Input Modalities

While the previous parts of the laboratory study were focused on output modalities the main research question of this part was to evaluate participants' opinion on the usage of different input modalities. Another aim was explore new ideas for rule-based input methods.

Procedure

This part of the laboratory lasted approximately 10 minutes. First each participant was shown a list of four predefined interaction rules as conceptualized in Section 3.3.9 (see Fig. 3.15). The rules cover four frequently used smartphone applications (camera, messaging, phone, maps) and all offered input possibilities (buttons, orientation, motion gestures, touch gestures). As touch and

Table 5.3.: Summarized comparison of the three rule awareness methods. The best results for each category are highlighted in blue. Statistically significant results are marked with a *.

	Notification	**Widget**	**Alert**
Efficiency Execution Time (Mean)*	9.9s	7.6s	2.1s
Effectiveness Success Rate* Number of errors per task (Avg.)	54.2% 0.08	87.5% 0.0	100% 0.04
Satisfaction Preference	41.7%	25.0%	20.8%

motion gesture recognition was not in scope of this thesis, we created a Wizard of Oz experiment so that the participants could try out the input methods by themselves: once the input gesture was performed as defined in the rule the according action (such as opening an app) was triggered from the instructors computer by sending a prepared message to the smartphone over the local wireless network. Thus the participant was given the feeling that she or he had triggered the action by her-/himself. After trying each rule the participants were asked to give feedback in a questionnaire. Lastly the participants were given the chance to create own input rules using the GUI for input modalities (see Fig. 3.13).

Results

The participants rated the given input rules on five-point Likert scales to evaluate how convenient, useful, socially acceptable and fun they appeared to them. All rules were accepted very well: taking the average value of all four rules 75% agreed that the input methods seem convenient, 74% agreed that they would use them (75% also in public) and 71% agreed that the input rules were fun to use. The results for each input rule are shown in Fig. 5.21. Two participants explicitly mentioned that the given input methods "feel natural". Another participant stated that he found the given rules convenient, because they were easy to remember. It was apparent that the motion gesture (*Move phone to ear and back*) was accepted least of all input methods. Potential reasons include that this gesture requires a relatively high motoric effort, also it is apparent from Fig. 5.21 that it was the socially least accepted: 41% would not use this gesture in public.

The participants created a great variety of different rules when being asked for further rules which would be useful to them. All available types of input options (orientation, touch gesture, motion gesture, buttons) were used. Suggested rules making use of the smartphone's **orientation** included

- turning the smartphone around so that the display is facing down to mute it ('"flip to mute"), and

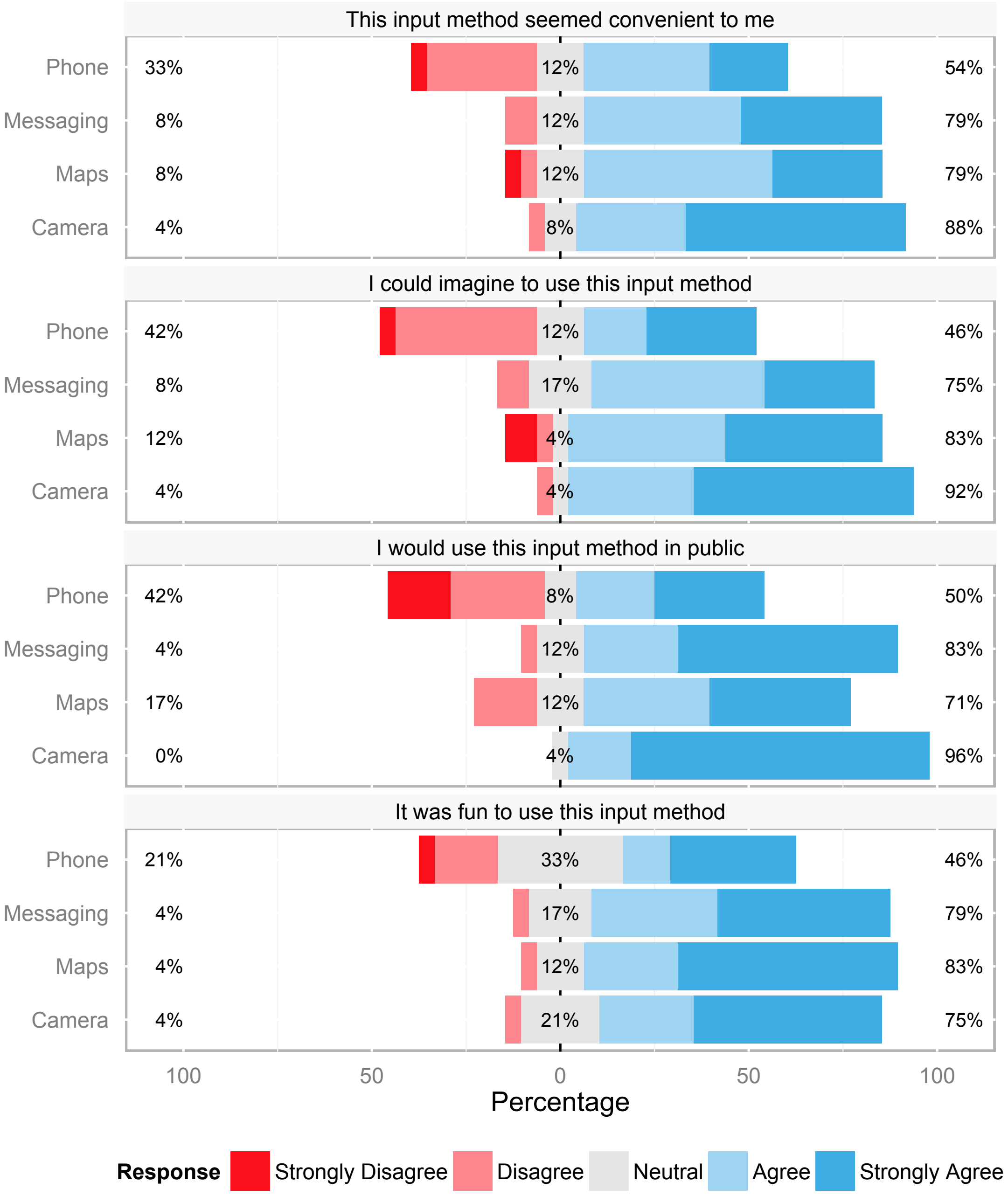


Figure 5.21.: Rating of the conceptualized input methods regarding convenience, usefulness, social acceptance and enjoyment.

- putting the smartphone upwards so that it is standing to open the watch app.

Touch gestures accounted for the greatest number of defined rules. Suggestions included

- performing a two-finger tap to unlock the smartphone (instead of a swipe gesture),
- swiping up- or downwards to adjust the smartphone's sound level,
- swiping sidewards to adjust the screen brightness,
- painting a checkmark to open the todo app
- painting a square to open the calendar app,
- painting a "N" to open the notes app,
- painting a "B" to open the web browser or other letters to open specific web sites.

One participant suggested a **motion gesture** described as

- quickly moving the phone downwards (letting it fall down while laying in the hand a short distance) to open the messaging app.

Most proposed **button** input rules can be summarized as assigning different functions to buttons depending on the current usage context. Propositions included

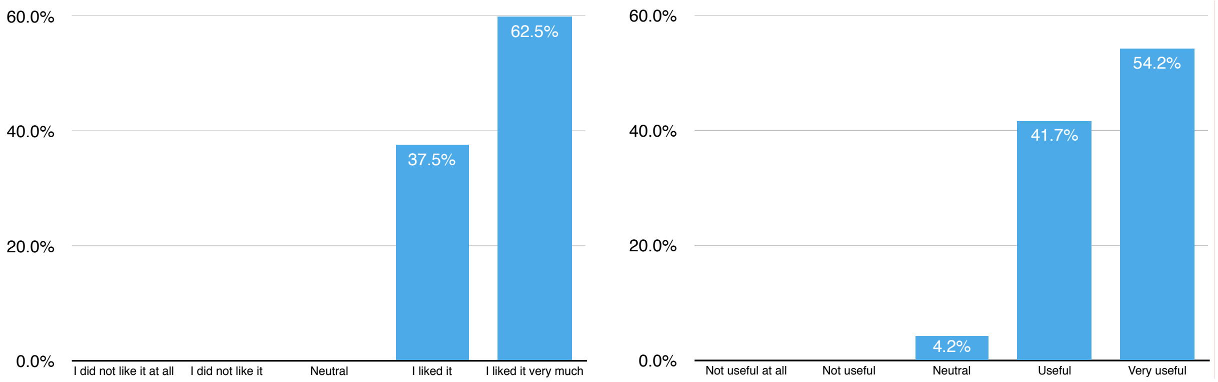
- using a button to disable screen rotation when using a news reading app, and
- using a button to en-/disable vibration when the phone is lying on a desk (the participant was unsatisfied with his smartphone's possibilities to adjust the vibration).

Taken together the user based creation of input rules appears to be an encouraging approach to a richer usage of input modalities. The variety of different rules proposed by the participants suggest that individual input methods are advantageous.

5.1.9. General Feedback

To gain general feedback on the proposed rule-based approach to multimodal interaction and the created user interfaces the participants were lastly given the chance to rate their overall experience with the application after participating in the laboratory study. All participants liked the app (62.5% liked it very much, see Fig. 5.22a). 95.8% considered the app useful (54.2% very useful, see Fig. 5.22b).

The general attractiveness of the application was measured using AttrakDiff [36], a simple six-item scale which aims to evaluate relevant aspects of the user experience. As evident from Fig. 5.23 the application offers a good user experience with regard to the AttrakDiff questionnaire.



(a) Percentage of participants who liked the application

(b) Percentage of participants who considered the application useful

Figure 5.22.: General Feedback on the application

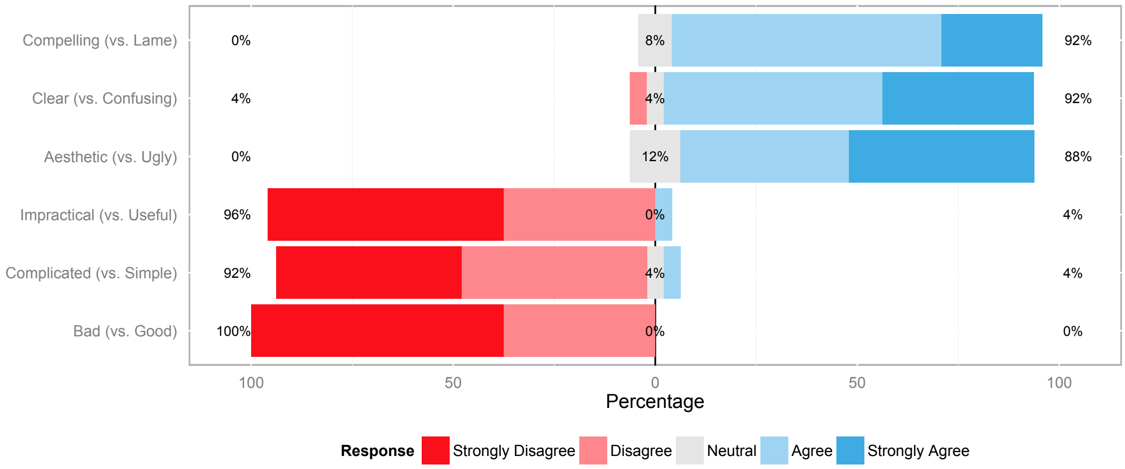


Figure 5.23.: Results of the AttrakDiff evaluation.

5.2. Field Study

In addition to the laboratory study a field study was conducted to gain feedback on the background of realistic usage situations where rules can be applied in everyday usage contexts.

5.2.1. Experiment Setup

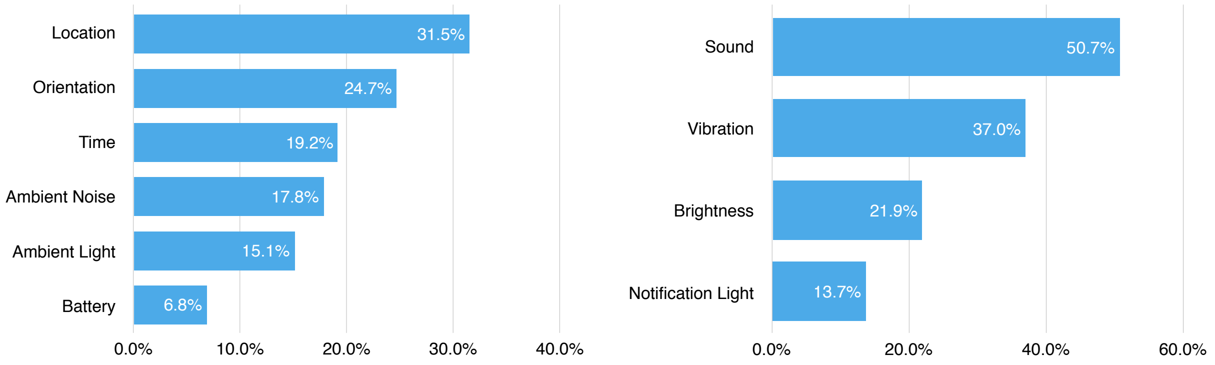
The participants were asked to use a fully functional Android application featuring the *Balanced* layout and no notification method on their own smartphones over a two week period. The used application was equipped with a self-written logging component to gather usage statistics over time. The participants were furthermore asked to install a questionnaire Android application named SERENA (Self-Reporting and Experience Sampling Assistant, [37]) on their devices, which reminded the participants to fill out a short questionnaire on their device every day. Thus it was possible to get feedback on errors which testers might forget after the two week period. After this period the participants were asked to submit the statistics data to a corresponding web server by tapping a button in the app and to fill out a conclusive online form to give feedback on their experience.

5.2.2. Participants & Demographic Data

For the field study five participants were recruited. Four of them were male, the average age was 26.6 years (range: 23-31 years, standard deviation 3.0 years). Three of the participants were students, one participant was a software developer and one participant was a physician. Three participants had a bachelor's degree, one participant reported a master's degree as the highest educational achievement, one participant had a PhD degree. They were recruited using mailing lists and social networks. All participants had a western background. The owned smartphone were three LG Nexus 5 devices, one Nexus 4 and one Sony Xperia Z1 Compact smartphone, all running Android 4.4. Four of the five participants rated their technical skill with relation to smartphones with 5, one participant with 3.

5.2.3. Results

One interesting research question behind the field study was to find out which contexts and modalities are particularly relevant to the users. The statistics gathered during the usage of the application reveals that on average almost one third of all rules contained location as a context (see Fig. 5.24a), which makes it the most frequent possibility to describe a situation. The orientation context was used in almost one quarter of all rules (24.7%), while the time context was the third most used context with (19.2%). This circumstance was also confirmed when asking



(a) Percentage of rules containing the different context types.

(b) Percentage of rules containing the different modalities.

Figure 5.24.: Usage of contexts and modalities in the field study.

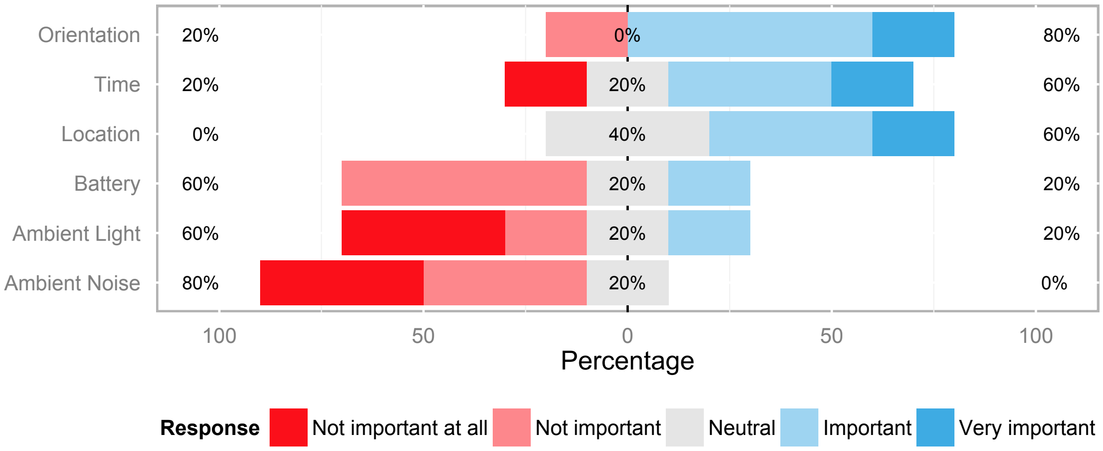


Figure 5.25.: Importance of the different contexts as rated by the participants of the field study.

the participants how important the different contexts were to them in the final questionnaire: orientation, time and location were rated as the most important contexts (see Fig. 5.25). It is apparent that the ambient noise context was used relatively frequently (in 17.8 % of all rules), but still rated as the most unimportant context. Possible explanations for this finding could be that the ambient noise context turned out to be less useful in actual situations than first expected or that the noise recognition did not work well. This was however not mentioned in the questionnaires by any participant.

Regarding modalities the results were similarly clear. Evaluating the usage statistics the sound modality was contained in more than half of all rules, the vibration modality was used in 37.0% (see Fig. 5.24b). These two modalities were also rated the most important in the questionnaire (see Fig. 5.26).

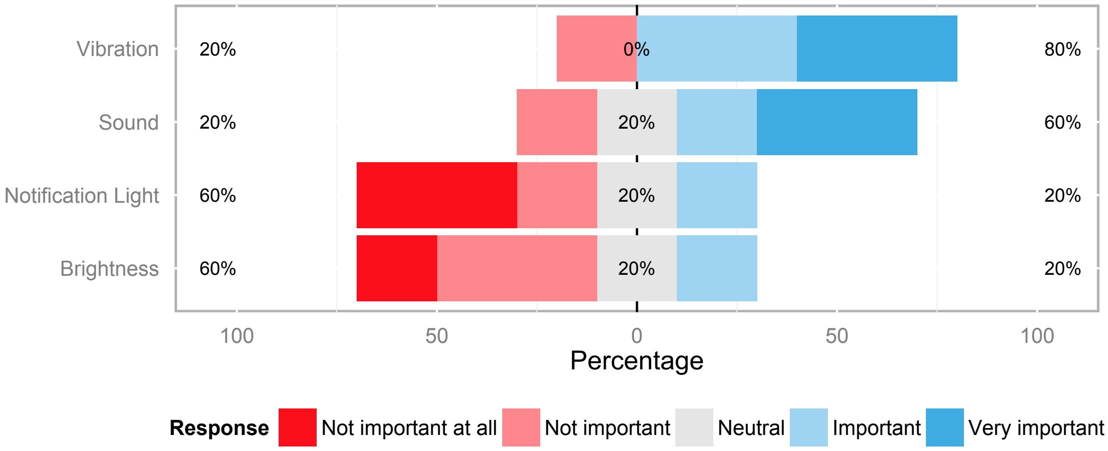
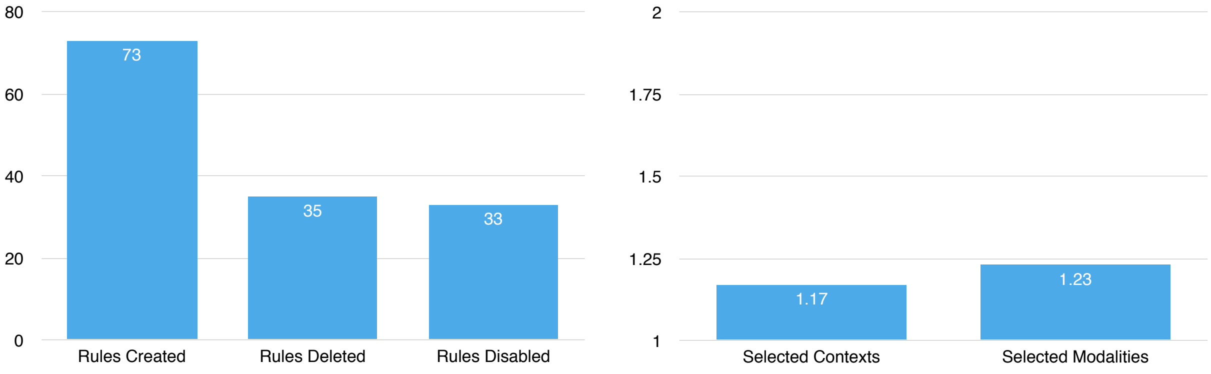


Figure 5.26.: Importance of the different modalities as rated by the participants of the field study.

The usage statistics furthermore collected data about the how many rules were created, deleted and disabled (see Fig. 5.27a). Taken together the participants created 73 rules which results in an average number of 14.6 created rules per user. The fact that almost one half of these rules (35 of 73) were deleted again after some time can possibly be explained by the behavior of experimenting with different kinds of rules to explore the capabilities of the application or by rules which have emerged as unprofitable for the user. The possibility to temporary disable rules appeared to be a well accepted feature: almost half of all created rules (33 of 73) were temporarily disabled. The collected data furthermore allows to draw conclusions about the common structure of created rules, meaning how many contexts options on the one hand and modalities on the other hand were commonly contained in a rule. Fig. 5.27b shows that most rules had a simple structure with an average of 1.15 selected contexts and 1.23 selected modalities. This confirms the expectation from Section 3.3.1 that most rules will be rather simple with only a small number groups and boolean connections.

The application used in the field study was build to apply modality changes in the background without notifying the user (this method was preferred by 12.5% of all participants in the laboratory study, see Fig. 5.20). When asking the participants of the field study whether they would have liked to get informed about modality changes in the final questionnaire the answers were very clear: all participants reported that they would like to get status feedback. Being asked about their preferred way to be notified the results were however similarly inconclusive as in the laboratory study: while 60% mentioned *Notifications* as their preferred method, 20% preferred a *Widget* while the remaining 20% argued in favor of the *Alert*.

Just like the participants of the laboratory study, all users of the field study application were asked to a answer a SUS questionnaire to get feedback about their overall satisfaction with the



(a) Number of rules created, deleted and disabled by the participants during the field study.

(b) Average number of contexts and modalities in rules created during the field study.

Figure 5.27.: Usage behavior regarding the creation, deletion and editing of rules as well as the structure of created rules.

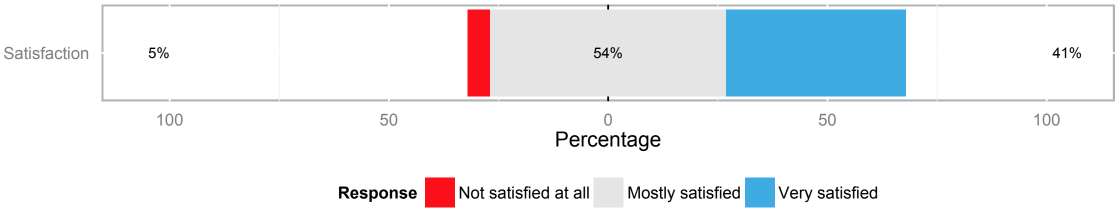


Figure 5.28.: Satisfaction of the field study participants as rated in the daily questionnaire.

application. With a SUS score of 75.0 the application provides a good satisfaction according to [35] (good: between 71.4 and 85.5, perfect: higher than 85.5).

The daily questionnaire furthermore allowed to collect feedback about the participants satisfaction with the application and its reliability on a daily basis using three-point Likert scales. Summarized the participants rated their satisfaction with the application as “not satisfied at all” in 5% of all answers while 54% of all answers were “mostly satisfied” and 41% were “very satisfied” (see Fig. 5.28). The reliability was rated similarly with 3% of all answers rating the application as “not reliable at all” while 46% of all answers were “mostly reliable” and 51% were “very reliable” (see Fig. 5.29). Thus, the experience regarding satisfaction and reliability was largely positive. Another feedback component of the daily questionnaire was a free text field asking for problems or anything that frustrated the user throughout a day. Most concerns referred to negative influences on the smartphone’s battery life (e.g. “drains too much battery”). Two participants requested improvements for the location context dialog such as defining a custom radius for location pins and the possibility to search for locations in a text field.

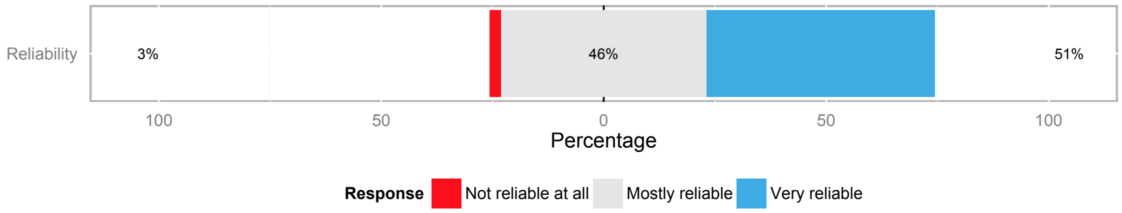


Figure 5.29.: Reliability of the application as rated in the daily questionnaire by the field study participants.

5.3. Discussion

Both user studies have revealed interesting insights about users' current modality usage, significant results about the best way to create rules and to notify the user about of modality changes, as well as findings about the long term experience with the proposed rule-based approach. The exploratory parts of the the conducted studies helped to evaluate how well users adopt the concept of multimodal interaction regarding both out- and input modalities.

It was interesting to see that almost half of the participants (42%) assessed their smartphone's general mode with regard to modalities as inappropriate. This appears even more adverse on the background that users seem to spend a lot of time manually changing modalities (96% of all participants reported to manually change their mobile device's modalities on a daily basis, 21% at least five times a day). The findings suggest that the prevalent ways to adjust modalities to different situations might not be optimal for a big part of smartphone users. A rule-based system can help to reduce the user's task load in this regard. There are two inevitable prerequisites for a system that adjusts modalities based on the current usage context, which were both accepted by the great majority of all participants: over 87% of all participants reported that it would be acceptable for them if their smartphone gathered context information in the background, over 95% reported that they would appreciate if their smartphone automatically adjusts their device's modalities.

The evaluation of different user interfaces for the creation of rules resulted in a clear winner: the *Situation-oriented* layout performed significantly better than the two other layouts regarding efficiency and satisfaction (regarding effectiveness all three layouts showed high success rates of over 85%, the differences among them being statistically non-relevant). Theses results show that the users' mental model is tendentially oriented towards first defining situations and then assigning suitable modalities to them in the second step (instead of assigning situations to separate modality sets as dictated in the *Modality-oriented* layout). Another difference of this layout in comparison to the other two variants lies in the usage of simple toggle buttons which allow to enable or disable each modality with one tap (instead of adding desired modalities triggers to a list as in the *Balanced* layout). In hindsight, this approach appears more reasonable when realizing the differences to selecting context factors: while selecting a series of context factors of the same type (e.g. location is X OR location is Y OR location is Z) is a common use case, this would not be sensible for modalities. Each modality (sound, vibration, notification LED etc.) can either be turned on or off. It makes no sense to define two triggers for the same modality type as they would conflict with each other, so that all modality options can be shown in a spatially limited area on the right side of the rule view.

The suggestion-based approach to rule creation was appreciated by almost 80% of all participants and can therefore be seen as an eligible addition to the manual rule creation process, which can help users to find rules for recurring situations they are not actively aware of. The available context options and the provided approach of logical AND and OR connections were assessed as adequate to describe relevant situations by all participants. Evaluating the relevance of different context factors, the current user location has turned out to be the most important way to describe changing situations. On the modality side, sound and vibration seem to be the most important settings. The GUI of a mobile application for rule-based interaction can incorporate these findings by arranging context factors an modalities according to the found usage frequency.

The most common structure of created rules appears to be very simple: both, rules suggested in the laboratory study and rules created throughout the field study mostly contain only one or two context options and modalities. Optimizing the GUI for simple scenarios by introducing some limitations has shown to be a useful step for this reason. The statistics gathered during the field study have furthermore demonstrated that the ability to temporarily disable particular rules is an appreciated feature.

The preferred notification type to provide awareness about modality changes has shown to be very subjective from person to person. While the *Notification* awareness method did not score best with regard to efficiency and effectiveness it was the method with the highest acceptance (preferred by more than 40% of all participants). Still, in both the laboratory and field study there was also a high percentage of users favoring the awareness methods *Widget* and *Alert*. The level of obtrusiveness has shown to be the most determining factor here. While obtrusive awareness methods like the *Alert* were found appropriate for users which desire a high level of control, other users preferred not to be notified at all or *Notifications* because of a low level of obtrusiveness. Taken together these findings suggest that a smartphone application for rule-based multimodal interaction should offer user settings with the possibility to select a preferred awareness method to take the subjective preferences into account. Having the highest agreement level (over 40%) the *Notification* can be seen as a suitable default setting.

All conceptualized rules for new input methods were accepted very well. Participants mentioned that the given input methods “feel natural” and stated that they feel convenient and easy to remember. Taken together the user based creation of input rules appears to be an encouraging approach to a richer usage of input modalities. Also the variety of different rules proposed by the participants suggest that individual input methods are advantageous. All in all the rule-based approach was appreciated very well: all participants liked the app (62.5% liked it very much), 95.8% considered the app useful (54.2% very useful). Low battery consumption and high reliability have emerged as the most important requirements of a useful system.

Chapter 6.

Conclusion

6.1. Summary

Modern mobile devices offer an increasing variety of in- and output hardware components which were described with regard to their properties, specific advantages and disadvantages and classified by the modalities they address. We communicate *visually* through screens, LEDs and cameras, *aurally* through speakers and microphones and *haptically* through vibration, touchscreens and buttons. While visual modalities mostly require the user's direct focus but allow to transfer highly specific information, auditory and tactile modalities can generally be perceived without direct focus but tendentially only convey a limited amount of information. Multimodality bears the chance to overcome perceptual or cognitive limitations and impaired motor skills experienced in varying situations. Combining multiple modalities can result in higher efficiency and improved error handling and allows for more natural interaction as well as more flexibility and personalization.

Modern mobile devices are also equipped with an increasing amount of sensors, which allow the collection of information characterizing the current usage situation. The relatively abstract notion of context information can be divided in two main categories and further subcategories to gain an overview of more concrete context factors. While conditions such as ambient light, noise, time or (device) orientation and a user's (rough, precise or moving) location can be summarized as context information about the *physical environment*, the user's activity (habits, emotional state) and social environment can be seen as *human factors* that provide context information.

Research in the area of multimodal interaction has mainly focused on modalities per se, but less on concepts which allow the user to control these in a beneficial way. Different approaches for systems supporting multimodal interaction are conceivable. They can be characterized along three dimensions: *system autonomy*, *user task load* and the *risk of unwanted modality changes*. While the current situation of manual modality switching results in a high task load for the user, a fully automated system without a GUI can result in unwanted modality changes combined with insufficient transparency for the user ("black box" behavior). The presented *rule-based* approach combined with the created GUIs to define these rules is positioned between these two extremes and aims to reduce the user's task load while leaving control about modality changes to the smartphone owner. A focus group helped to identify basic requirements and to generate initial ideas for a rule-based system supporting multimodal interaction. Even in a small group a diverse usage of modalities could be observed. Unsatisfactory methods to adjust modalities can result in a single standard setting as a suboptimal "lowest common denominator" for varying situations. Reliability and providing awareness emerged as important features of a system that controls multimodal interaction.

Graphical user interface (GUIs) for all parts of a rule-based application were created. Firstly, a process to represent and define situations was designed. The design is optimized for the most common scenarios and avoids situation structures that could be ambiguous from a user's perspective. Context factors and modalities can be defined in dialogs which present the user an own view for each task without completely leaving the current context of the application's main view. One interesting question that emerged during the concept phase, was whether persons think about context-modality combinations in either a *situation-oriented*, a *modality-oriented* or a *balanced* way. GUIs for each variant were created in order to later identify the best ways to create rules. Additionally GUIs for alternative ways to create a useful set of rules were conceived. An application could provide *suggestions* based on a machine-learning approach to the user by registering if certain modalities are frequently used in particular situations. Also different methods to provide awareness about modality changes were conceived. The different possibilities: *Widget*, *Alert* and *Notification*, can be distinguished along the two dimensions *obtrusiveness* and *control*.

The created GUIs also allow the user to define rules combining different *input modalities* to evoke actions such as opening an application. In contrast to output modalities which can be adapted to changing contexts in a meaningful way, it is generally not necessary to disable certain *input* modalities in a particular situation. Still it is useful to have different input modalities for the same task to offer suitable input methods in varying contexts. We came up with the idea of "mimicry gestures" which mimic natural behavior of people using their smartphone to trigger common tasks. Besides being socially acceptable through their hardly noticeable appeal, we see possible advantages in these input methods in mentally preparing the user for the task to be evoked and the fluid transition to the actual task.

The conceptualized GUIs and the rule-based system were realized as an Android application which allows users to define rules for context-sensitive output modality switches and mulitmodal input methods. After extending the underlying *M3I* framework with additional context factors and triggers, as well as the possibility to maintain activity states for each rule in the evaluator, a structure of GUIs which can easily be extended by application developers was created. Graphical representations of situations created with these GUIs can be translated to logical expressions with the help of a demonstrated algorithm. Furthermore a system consisting of broadcasts, broadcast receivers and services which allows to observe context changes and to adapt modalities in the background was presented.

The developed GUIs and Android application were evaluated in a laboratory and a field study. The *situation-oriented* rule layout could be identified as the most efficient and satisfactory user interface to create rules, which indicates that the users' mental model is tendentially oriented towards first defining situations and then assigning suitable modalities to them in a second step. The *suggestion*-based approach to rule creation was well appreciated and can therefore be seen as an eligible addition to the manual rule creation process. The preferred method to provide *awareness* about modality changes has shown to be very subjective from person to person. The findings suggest that a mobile application for rule-based multimodal interaction should offer user settings with the possibility to select a preferred awareness method to take these subjective preferences into account. All conceptualized rules for "mimicry" *input* methods were accepted very well. Participants mentioned that the given input methods "feel natural" and stated that they feel convenient, socially acceptable and easy to remember.

Taken together the user based creation of output and input rules was liked and considered useful by the large majority of all users and appears to be an encouraging approach to a richer usage of modalities. We hope that the created GUIs can serve as a guideline for designers and that our findings about users' preferences, opinions and behavior regarding multimodal interaction as well as the demonstrated implementation can provide useful information to researchers and developers in the area of mobile multimodal applications.

6.2. Future Work

Even though all participants of the laboratory study affirmed that they could sufficiently describe relevant situations with the offered set of context factors, it would be interesting to implement and investigate more methods to describe a user's context. Wearable activity trackers that measure a person's physiological parameters (e.g. the Fitbit Tracker [1]) could be used to describe a person's current activity. This would for example allow to disable obtrusive modalities when the smartphone registered that a person fell asleep. Also more exact localization technologies such as BLE or NFC could be used to determine a person's presence in particular rooms (e.g. in the living room) or proximity to certain objects (e.g. mobile device is placed next to the bed). It is expectable that mobile devices will be equipped with even more sensors in the future. The capability to process more context information could help to describe particular situations more precisely and thus, to avoid unintended modality changes more effectively.

The *suggestion*-based approach to automated rule creation and the necessary machine learning should be further investigated as a suitable addition to the manual definition of rules. A challenge

[1] http://fitbit.com/

which might emerge in this area is to determine those context factors which are relevant to describe a recurring situation and those who are not.

As described above, the conceptualized input rules (*mimicry gestures*) were well appreciated in the conducted Wizard of Oz experiment. It would be interesting to complete the implementation of the necessary gesture recognition in the *M3I* framework and Android application to further investigate the gestures' practical utility in another field study.

List of Figures

2.1. Modalities described as in- and output communication channels in HCI. 8

2.2. Smartphone with a notification LED integrated in the earpiece area. Image from: [12] 9

2.3. Coordinate system (relative to a mobile device) that is used for the accelerometer, magnetometer and gyroscope sensors by the Android Sensor API. Image from: [20]. 13

2.4. Sensor board developed by Schmidt et al. [23] and Context Studio by Korpipää et al. [24]. 17

2.5. Location-aware mobile application *LLama*. 19

3.1. Illustration of the relation between a user's task load, the autonomy of a system and the risk of unwanted modality changes for context-aware modality switching. 21

3.2. Examples of common situations as representations in the GUI. 25

3.3. Situation structures and layouts to be avoided. 26

3.4. Different approaches to creating an expression. 28

3.5. GUIs created to define context factors. 29

3.6. GUIs created to define output modality switches. 30

3.7. Selection of icons which were used to clarify the meaning of GUI elements. . . . 31

3.8. Rule layout variants corresponding to different user mind sets which emerged during the concept phase. 32

3.9. Navigation flow when creating a rule (using the *Balanced* layout). 33

3.10. Three awareness methods: *widget*, *notification* and *alert*, positioned in comparison to each other regarding obtrusiveness and control. 34

3.11. Overview about different methods to provide awareness about modality changes to the user. 35

3.12. Rule creation alternatives: *Suggestion* & *Snapshot*. 37

3.13. GUIs created for the defintion of input rules. 38

3.14. Typical posture when taking a photo (image from http://www.huawei-university.at). 39

3.15. Conceptualized input rules as defined in the GUI. 40

4.1. General structure and components of the *M3I Framework for Context-Based Mobile Multimodal Interaction*, which provides the basis for the Android application's implementation. Drawing based on [33]. 43

4.2. Extended version of the Evaluator. Maintaining active/inactive states for each rule and saving current modality settings as elseTriggers allows to return to the previous modality settings once a rule is first evaluated false. An enabled/disabled state allows to temporarily disable certain rules. 46
4.3. Main user interfaces of the developed Android application. 49
4.4. Different states of the RuleActivity. 50
4.5. Graphical representation (left) with the equivalent logical expression (right). . . . 53
4.6. BroadcastReceivers and Services running in the background of the application to ensure and optimize the continuous evaluation of rules. 56
4.7. Web application developed for the conduction of the laboratory study to remotely trigger *Notifications*, *Alerts*, *Widget* updates and applications (camera, messaging, maps and phone) on the test device. 57

5.1. Distribution of participants' age and educational degrees. 60
5.2. Currently owned smartphone by manufacturer (left) and self-estimated technical expertise (right, the bold black line in the right diagram shows the median. . . . 60
5.3. Modality changes made on a regular basis. 61
5.4. Frequency of modality changes on an average day. 62
5.5. Ways to change modalities. 62
5.6. General mode with regard to in/output modalities. 63
5.7. Situations in which participants experience inappropriate modalities. 63
5.8. Acceptance of the smartphone gathering context information via sensors in the background. 64
5.9. Acceptance of the smartphone automatically adjusting modalities. 64
5.10. Acceptance of different rule scenarios. 65
5.11. Steps of each rule creation task, exemplary shown for a task with layout *Balanced* and rule type *1*. 66
5.12. Comparison of the three layout variants *Situation-oriented*, *Modality-oriented*, and *Balanced* by execution time. The bold black line in each bar shows the median. . 68
5.13. Comparison of the three layout variants *Balanced*, *Modality-oriented* and *Situation-oriented* by success rate and average number of errors made by the participants. 69
5.14. Comparison of the three layout variants *Situation-oriented*, *Modality-oriented*, and *Balanced* by their results in the SUS questionnaire. 71
5.15. Acceptance of the rule creation alternative *Suggestion* 72
5.16. Acceptance of the rule creation alternative *Snapshot* 73
5.17. Common useful rules as named by participants. 74
5.18. Preferred ways of the smartphone to react when a user leaves a defined context. 75
5.19. Comparison of the three notification types Notification, Widget and Alert by execution time (ms). The bold black line in each bar shows the median. 76

5.20. Preferred ways to get informed about modality changes 78
5.21. Rating of the conceptualized input methods regarding convenience, usefulness, social acceptance and enjoyment. 80
5.22. General Feedback on the application 82
5.23. Results of the AttrakDiff evaluation. 82
5.24. Usage of contexts and modalities in the field study. 84
5.25. Importance of the different contexts as rated by the participants of the field study. 84
5.26. Importance of the different modalities as rated by the participants of the field study. 85
5.27. Usage behavior regarding the creation, deletion and editing of rules as well as the structure of created rules. 86
5.28. Satisfaction of the field study participants as rated in the daily questionnaire. . . 86
5.29. Reliability of the application as rated in the daily questionnaire by the field study participants. 87

List of Tables

1.1. Modalities and modality combinations best suited for different situations [5]. . . 3

2.1. Summary of in- and output devices of modern smartphones classified by used modalities. 11
2.2. Overview about different categories of context information based on the context feature space by Schmidt et al. [17]. 15

5.1. Rule types chosen for the rule creation task of the laboratory study. The underlined entries indicate the layout which shows an advantage when creating the given rule type. 67
5.2. Summarized comparison of the three evaluated layouts. The best results for each category are highlighted in blue. Measurements with statistically significant results are marked with a *. 70
5.3. Summarized comparison of the three rule awareness methods. The best results for each category are highlighted in blue. Statistically significant results are marked with a *. 79

List of Acronyms

HCI	Human-Computer Interaction
GUI	Graphical User Interface
HMI	Human Machine Interface
SUS	System Usability Scale
GNSS	Global Navigation Satellite System
BLE	Bluetooth Low Energy
NFC	Near Field Communications
GCM	Google Cloud Messaging
MVC	Model-View-Controller
ANOVA	Analysis of Variance
IEEE	Institute of Electrical and Electronics Engineers
TUM	Technische Universität München
VMI	Fachgebiet Verteilte Multimodale Informationsverarbeitung

Bibliography

[1] D. Ferreira, *AWARE: A Mobile Context Instrumentation Middleware To Collaboratively Understand Human Behavior.* PhD thesis, University of Oulu, 2013.

[2] S. Tamminen, A. Oulasvirta, K. Toiskallio, and A. Kankainen, "Understanding Mobile Contexts," *Personal and Ubiquitous Computing*, vol. 8, no. 2, pp. 135–143, 2004.

[3] D. Hakkani-Tur, G. Tur, and L. Heck, "Research Challenges and Opportunities in Mobile Applications," *Signal Processing Magazine, IEEE*, vol. 28, no. 4, pp. 108–110, 2011.

[4] L. Chittaro, "Distinctive Aspects of Mobile Interaction and their Implications for the Design of Multimodal Interfaces," *Journal on Multimodal User Interfaces*, vol. 3, no. 3, pp. 157–165, 2010.

[5] S. Lemmelä, A. Vetek, K. Mäkelä, and D. Trendafilov, "Designing and Evaluating Multimodal Interaction for Mobile Contexts," in *Proceedings of the 13th Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting, Liverpool*, pp. 265–272, ACM, 2008.

[6] S. Oviatt, "Multimodal Interfaces for Dynamic Interactive Maps," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 95–102, ACM, 1996.

[7] F. Quek, D. McNeill, R. Bryll, S. Duncan, X.-F. Ma, C. Kirbas, K. E. McCullough, and R. Ansari, "Multimodal Human Discourse: Gesture and Speech," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 9, no. 3, pp. 171–193, 2002.

[8] M. Turk, "Multimodal Interaction: A Review," *Pattern Recognition Letters*, vol. 36, no. 1, pp. 189–195, 2013.

[9] B. Dumas, D. Lalanne, and S. Oviatt, "Multimodal Interfaces: A Survey of Principles, Models and Frameworks," in *Lecture Notes in Computer Science*, pp. 3–26, Springer Berlin Heidelberg, 2009.

[10] S. Oviatt and P. Cohen, "Perceptual User Interfaces: Multimodal Interfaces that Process What Comes Naturally," *Communications of the ACM*, vol. 43, no. 3, pp. 45–53, 2000.

[11] A. K. Dey, G. D. Abowd, and D. Salber, "A Conceptual Framework and a Toolkit for Support-

ing the Rapid Prototyping of Context-Aware Applications," *Human-Computer Interaction*, vol. 16, no. 2, pp. 97–166, 2001.

[12] K. Bell, "Moto X's Hidden Notification LED Can Be Enabled With Root Access. Available: http://www.cultofandroid.com/49383/moto-xs-hidden-notification-led-can-enabled-root-access-video/," 2014.

[13] A. van Dam, "Post-WIMP User Interfaces," *Communications of the ACM*, vol. 40, no. 2, pp. 63–67, 1997.

[14] R. J. K. Jacob, A. Girouard, L. M. Hirshfield, M. S. Horn, O. Shaer, E. T. Solovey, and J. Zigelbaum, "Reality-Based Interaction: A Framework for Post-WIMP Interfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 201–210, ACM, 2008.

[15] A. Sinhal, A. Chaturvedi, and V. Khan, "Human Computer Interaction: Overview on State of the Art," *International Journal on Smart Sensing and Intelligent Systems*, vol. 1, no. 1, pp. 137–159, 2008.

[16] J. R. Williamson, *User Experience, Performance, and Social Acceptability: Usable Multimodal Mobile Interaction*. PhD thesis, University of Glasgow, 2012.

[17] A. Schmidt, M. Beigl, and H.-W. Gellersen, "There is more to Context than Location," *Computers & Graphics*, vol. 23, no. 6, pp. 893–901, 1999.

[18] S. Reddy, M. Mun, J. Burke, D. Estrin, M. Hansen, and M. Srivastava, "Using Mobile Phones to Determine Transportation Modes," *ACM Transactions on Sensor Networks (TOSN)*, vol. 6, no. 2, pp. 13–27, 2010.

[19] S. Ayub, A. Bahraminisaab, and B. Honary, "A Sensor Fusion Method for Smart phone Orientation Estimation," in *13th Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting*, 2012.

[20] Google Inc., "Android Sensors Overview. Available: http://developer.android.com/guide/topics/sensors/sensors_overview.html," 2014.

[21] R. Want, A. Hopper, V. Falcão, and J. Gibbons, "The Active Badge Location System," *ACM Transactions on Information Systems (TOIS)*, vol. 10, no. 1, pp. 91–102, 1992.

[22] B. Schilit, N. Adams, and R. Want, "Context-Aware Computing Applications," in *Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications*, pp. 85–90, IEEE, 1994.

[23] A. Schmidt, K. A. Aidoo, A. Takaluoma, U. Tuomela, K. Van Laerhoven, and W. Van de

Velde, “Advanced Interaction in Context,” in *Proceedings of the 1st International Symposium on Handheld and Ubiquitous Computing*, pp. 89–101, 1999.

[24] P. Korpipaa, J. Hakkila, J. Kela, S. Ronkainen, and I. Kansala, “Utilising Context Ontology in Mobile Device Application Personalisation,” in *Proceedings of the rd International Conference on Mobile and Ubiquitous Multimedia*, pp. 133–140, ACM, 2004.

[25] D. Siewiorek, A. Krause, A. Smailagic, J. Furukawa, N. Moraveji, A. Krause, N. Moraveji, K. Reiger, J. Shaffer, and F. L. Wong, “Sensay: A Context-Aware Mobile Phone,” in *Proceedings of the 13th Annual Post Graduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting, Liverpool*, p. 248, 2003.

[26] M. Raento, A. Oulasvirta, R. Petit, and H. Toivonen, “ContextPhone: A Prototyping Platform for Context-Aware Mobile Applications,” *IEEE Pervasive Computing*, vol. 4, no. 2, pp. 51–59, 2005.

[27] J. Froehlich, M. Y. Chen, S. Consolvo, B. Harrison, and J. A. Landay, “MyExperience: A System for In situ Tracing and Capturing of User Feedback on Mobile Phones,” in *Proceedings of the 5th International Conference on Mobile Systems, Applications and Services*, pp. 57–70, ACM, 2007.

[28] A. Möller, S. Diewald, L. Roalter, and M. Kranz, “MobiMed: Comparing Object Identification Techniques on Smartphones,” in *Proceedings of the 7th Nordic Conference on Human-Computer Interaction: Making Sense Through Design*, pp. 31–40, ACM, 2012.

[29] A. Möller, C. Kray, L. Roalter, S. Diewald, R. Huitl, and M. Kranz, “Tool Support for Prototyping Interfaces for Vision-Based Indoor Navigation,” in *Proceedings of the Workshop on Mobile Vision and HCI (MobiVis). Held in Conjunction with Mobile HCI*, pp. 1–10, 2012.

[30] Google Inc., “Project Tango. Available: `https://www.google.com/atap/projecttango/`,” 2014.

[31] J. Ruiz, Y. Li, and E. Lank, “User-Defined Motion Gestures for Mobile Interaction,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 197–206, ACM, 2011.

[32] J. Rico and S. Brewster, “Usable Gestures for Mobile Interfaces: Evaluating Social Acceptability,” in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 887–896, ACM, 2010.

[33] A. Möller, S. Diewald, L. Roalter, and M. Kranz, “M3I: A Framework for Mobile Multimodal Interaction,” *Mensch & Computer 2014: interaktiv unterwegs*, 2014.

[34] J. Brooke, “SUS - A Quick and Dirty Usability Scale,” *Usability Evaluation in Industry*, vol. 189, p. 194, 1996.

[35] A. Bangor, P. Kortum, and J. Miller, "Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale," *Journal of Usability Studies*, vol. 4, no. 3, pp. 114–123, 2009.

[36] M. Hassenzahl, M. Burmester, F. Koller, and H. Brau, "Der User Experience (UX) auf der Spur: Zum Einsatz von www. attrakdiff. de," *Usability Professionals*, pp. 78–82, 2008.

[37] A. Möller, M. Kranz, B. Schmid, L. Roalter, and S. Diewald, "Investigating Self-Reporting Behavior in Long-Term Studies," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 2931–2940, ACM, 2013.