Technische Universität München
Department of Electrical Engineering and Information Technology
Chair of Media Technology

Leveraging Mobile Interaction with Multimodal and Sensor-Driven User Interfaces

Dipl.-Medieninf. (Univ.) Andreas Möller

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

genehmigten Dissertation.

Vorsitzender: Univ.-Prof. Dr.-Ing. Wolfgang Kellerer

Prüfer der Dissertation:
1. Univ.-Prof. Dr. rer. nat. Matthias Kranz (Universität Passau)
2. Univ.-Prof. Dr. rer. nat. Uwe Baumgarten
3. Univ.-Prof. Dr.-Ing. habil. Gerhard Rigoll

Die Dissertation wurde am 06.11.2014 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 31.05.2015 angenommen.

Abstract

The increasing functionality of mobile devices and operating systems often entails an increment in complexity and complicatedness. This problem takes on greater significance with the opening towards new application areas (e.g., health and fitness) and new user groups (e.g., technically unversed people and the elderly). Interaction channels or modalities play here a central role: The increasingly pervasive use (in the context of Ubiquitous Computing) requires a stronger adaptation of these channels to changing usage contexts in order to ensure optimal interaction.

This work investigates multimodality as an approach to leverage the user experience with mobile devices. The use of multimodality is motivated by the numerous advantages identified in prior work, such as naturalness, efficiency, robustness, and popularity with users. The design space of multimodal interaction, as well as comprehensive support of multimodality from scratch in the development process has not been investigated so far in a holistic way for mobile devices.

The central research question of this dissertation is how to make multimodality usable to achieve better mobile interaction. This includes ease of operation and usability in existing scenarios, as well as opening up entirely new use cases.

The dissertation focuses on two aspects. First, an improvement from a user-centric point of view shall be achieved (measurable by, e.g., efficiency, error rate, and usability metrics). Therefore, the use of selected modalities and interaction methods is outlined in exemplary use cases, leading to a profound understanding of multimodality and its advantages in heterogeneous application areas. Second, from a developer-centric point of view, the implementation of multimodal interaction methods shall be simplified, thereby stronger motivating the consideration of multimodality in application development. To this end, a rule-based model and a software framework is presented, which supports multimodal in- and output and makes it usable for application programming. Moreover, approaches for end users to define multimodal behavior, as well as awareness on active modalities, are presented and evaluated. Beyond that, the dissertation exposes suitable evaluation methods for multimodal systems and points out characteristics of multimodal systems to be respected. The work thereby highlights and discusses all fundamental steps of the software development process, from design, prototyping, implementation to evaluation, with regard to mobile multimodal interaction.

Kurzfassung

Mit der zunehmenden Funktionsvielfalt mobiler Endgeräte und Betriebssysteme geht oft auch eine Erschwerung der Bedienung einher. Dieses Problem gewinnt an Bedeutung durch die Tatsache, dass fortwährend neue Anwendungsszenarien (z.B. der Gesundheits- und Fitnessbereich) und neue Zielgruppen (z.B. Senioren und technisch weniger versierte Nutzer) für mobile Interaktion erschlossen werden. Eine zentrale Rolle kommt hierbei den verschiedenen Interaktionskanälen bzw. -modalitäten zu. Die zunehmend allgegenwärtige Nutzung (Stichwort "Ubiquitous Computing") erfordert eine stärkere Adaption dieser Kanäle an wechselnde Nutzungsgegebenheiten und Kontexte, um optimale Interaktion zu gewährleisten.

In dieser Arbeit wird Multimodalität als Lösungsansatz untersucht, um das Benutzungserlebnis auf mobilen Geräten zu verbessern. Der Einsatz von Multimodalität ist motiviert durch zahlreiche Vorteile, die in vorangegangenen Arbeiten identifiziert wurden, z.B. Natürlichkeit der Interaktion, Effizienz, Robustheit und Beliebtheit bei den Anwendern. Der "Design Space" für multimodale Interaktion sowie eine umfassende Unterstützung von Multimodalität bereits im Entwicklungsprozess wurde jedoch bisher noch nicht ganzheitlich für mobile Geräte betrachtet.

Die zentrale Forschungsfrage dieser Dissertation ist, wie Multimodalität so nutzbar gemacht werden kann, dass eine bessere mobile Interaktion ermöglicht wird. Ziel dabei ist es, sowohl den Komfort und die Benutzerfreundlichkeit in existierenden Szenarien zu verbessern als auch völlig neue Szenarien und Anwendungsfelder zu erschließen.

Dabei setzt die Arbeit zwei Schwerpunkte: Zum ersten soll eine Verbesserung aus Anwendersicht erreicht werden (messbar z.B. durch Effizienz, Fehlerrate, und Usability-Metriken). Dazu wird der Einsatz ausgewählter Modalitäten und Interaktionsmethoden in exemplarischen Anwendungsbereichen aufgezeigt, was zu einem tiefergehenden Verständnis von Multimodalität und ihrer Vorteile in heterogenen Gebieten führt. Zum zweiten soll aus Entwicklersicht die Implementierung multimodaler Interaktionsmethoden vereinfacht und damit das In-Betracht-Ziehen von Multimodalität bei der Anwendungsentwicklung stärker motiviert werden. Hierzu wird ein regelbasiertes Modell sowie ein Software-Framework vorgestellt, welches multimodale Ein- und Ausgabe unterstützt und für die Programmierung eigener Anwendungen nutzbar macht. Des Weiteren werden Wege zur Festlegung multimodalen Verhaltens durch den Endanwender sowie des Bewusstseins über aktivierte Modalitäten vorgestellt und evaluiert. Die Dissertation legt darüber hinaus geeignete Methoden zur Evaluation multimodaler Systeme dar, und weist auf dabei zu beachtende Besonderheiten hin. Die Arbeit diskutiert und beleuchtet damit alle wesentlichen Schritte im Software-Entwicklungsprozess von Design, Prototyping, Implementierung bis Evaluierung im Hinblick auf mobile multimodale Interaktion.

Preface

This dissertation is based on approximately four years of research during my time as research assistant at the Institute for Media Technology at Technische Universität München (TUM). My work was situated in the Distributed Multimodal Information Processing Group (German: VMI, Verteilte Multimodale Informationsverarbeitung), and, since March 2013, in collaboration with the Embedded Interactive Systems Laboratory (EISLab) at the University of Passau.

The individual research was conducted in the course of different research projects, and, for some parts, in collaboration with other research groups at different institutes, namely:

- Culture Lab, University of Newcastle
- Sprachraum, Ludwig-Maximilians Universität München (LMU)
- Carl-von-Linde-Akademie, Technische Universität München (TUM)
- Teaching and Learning in Higher Education, University of Göttingen
- Institute for Geoinformatics, University of Münster
- Auto-Id Labs, ETH Zürich

Parts of the research that contributed to Chapter 5 were funded by the space agency of the German Aerospace Center with funds from the Federal Ministry of Economics and Technology on the basis of a resolution of the German Bundestag under the reference 50NA1107.

Excerpts of this work have already been published on international peer-reviewed conferences and workshops, and in international journals. The chapters of this dissertation are partly based on these publications. They are referenced at the beginning of the respective chapters. Furthermore, implementations and studies presented in this work have to some extent appeared in student theses I have supervised.

As a sign of appreciation for the support by everyone who helped to shape this work, in particular the co-authors of the above-mentioned publications, I will use the scientific plural in this dissertation.

The language used in this dissertation aims at gender neutrality. Whenever speaking of users or study participants in a generic sense, the pronouns *she* and *he* will be used alternately in an inclusive sense, in order to express that both females and males are meant.

References to the employed statistical tests are given at their first occurrence in the dissertation. The used symbols are listed for reference in the “Notation” section. Non-scientific references (e.g., web pages) will not be included in the bibliography at the end of the dissertation, but referred to in footnotes.

Acknowledgments

First and foremost, I would like express my gratitude to my supervisor, *Prof. Dr. Matthias Kranz*, who gave me the opportunity to pursue my research for this dissertation under his guidance. Not only did he introduce me to the joy (and the business) of science; being a professor at different universities (Munich, Luleå, and Passau) also never hindered him from being available any time for questions, discussions, feedback, and advice whenever necessary. I also thank *Prof. Dr. Uwe Baumgarten* and *Prof. Dr.-Ing. Gerhard Rigoll* for co-advising my thesis, and *Prof. Dr.-Ing. Wolfgang Kellerer* for being the head of the examination committee.

Big thanks go to my outstanding colleagues who made my time at TUM a delightful memory. Particularly, I want to acknowledge my office mates *Stefan Diewald* and *Luis Roalter* (for all the joint work efforts, support, and also the fun we had together). I further would like to mention the *NAVVIS* team, who was in parts involved in the indoor navigation chapter of this work. I also thank the *EISLAB* team at Passau (*Patrick Lindemann*, *Marion Koelle*, *Tobias Stockinger*) for the remote collaboration and the inspiring input.

This dissertation would not have been possible without the many cooperations that have lead to great research results and to numerous joint publications. I owe my gratitude to the activity recognition gurus *Dr. Thomas Plötz*, *Nils Hammerla* and *Prof. Dr. Patrick Olivier* at *Culture Lab*, and to *Dr. med. Johannes Scherr* for his professional medical advice, all contributing to the GymSkill project. For supporting the MobiDics project with their didactic expertise, my sincere appreciation goes to the *Sprachraum* team at LMU (especially *Dr. Barbara Meyer*, *Barbara Beege*, and *Dr. Andreas Hendrich*), as well as *Dr. Andreas Fleischmann*, *Angelika Thielsch*, and all other contributions to MobiDics. For their collaboration in joint publications, I thank *Prof. Dr. Chris Kray* (who co-authored a paper on indoor navigation) and *Dr. Florian Michahelles* (who co-authored a paper on research in the large).

I am very grateful that I had the opportunity to supervise many talented and committed students. I like to thank each of them who contributed to this dissertation with their diploma, bachelor, and master theses.

My final and most sincere thanks go to my girlfriend *Angelika* (who also proofread the thesis – special thanks for that!) and to my father, for always being there, standing behind me, encouraging and supporting me.

Contents

Notation

Abbreviations and Acronyms

AAL	Ambient Assisted Living
ADL	Activities of Daily Living
AJAX	Asynchronous JavaScript and XML
AoA	Angle of Arrival
API	Application Programming Interface
app	(Especially mobile) application
AR	Augmented Reality
CBIR	Content-Based Image Retrieval
CCD	Charge-Coupled Device
CMOS	Complementary Metal-Oxide Semiconductor
CSCC	Computer-Supported Coordinative Care
DECT	Digital Enhanced Cordless Telecommunications
DOF	Degrees Of Freedom
DPBN	Decision-Point Based Navigation
FAST	Features from Accelerated Segment Test
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GSM	Global System for Mobile Communications
GUI	Graphical User Interface
GymSkill	Gym Exercising Skill Assessment
HCI	Human-Computer Interaction
HMD	Head-Mounted Display
HRQ	High-Level Research Question
IEEE	Institute of Electrical and Electronics Engineers

IMU	Inertial Measurement Unit
IrDA	Infrared Data Association
LAN	Local Area Network
LED	Light-Emitting Diode
LIDAR	LIght Detection And Ranging
M3I	Mobile MultiModal Interaction
MIMO	Multiple Input Multiple Output
MobiDics	Mobile Didactics (application)
MobiMed	Mobile Medication Identifer (application)
MOOC	Massive Open Online Courses
MUSED	MUltimodal and SEnsor-Driven (relating to interaction or a user interface
NFC	Near Field Communication
PCA	Principal Component Analysis
PCBA	Principal Component Breakdown Analysis
PIL	Patient Information Leaflet
POI	Point of Interest
PZN	Pharmazentralnummer (german), a unique identification number for medications
QR	Quick Response
RFID	Radio Frequency IDentification
RQ	Research Question
SDK	Software Development Kit
SERENA	SElf-Reporting and ExperieNce sampling Assistant (application)
SUS	System Usability Scale
ToA	Time of Arrival
TDoA	Time Difference of Arrival
TUM	Technische Universität München
UbiComp	Ubiquitous Computing
UI	User Interface
VR	Virtual Reality
WLAN	Wireless Local Area Network
WOz	Wizard of Oz, according to [152]

Symbols

χ^2	Chi-squared value
df	Degrees of freedom
η^2	Effect size
F	F-test result
M	Mean
MD	Median
p	Probability
r	Pearson's product-moment correlation coefficient
SD	Standard deviation
W	Wilcoxon signed-rank and Mann-Whitney test statistic
Z	Standard score

Part I

Introduction and Background

Chapter 1

Introduction

1.1 Motivation

Smart mobile devices have become pervasive and an indispensable part of our lives. The abundance of available mobile applications (apps) has transformed mobile phones to all-purpose devices useful for the broad public, and the introduction of multi-touch interaction has revolutionized the interaction with these devices. The introduction of the iPhone[1] in 2007 is considered widely as a key trigger for this development. Smartphone and tablet devices have become more and more powerful and feature-rich ever since, and there is no end in sight for this development. However, an increment of options goes along with an increase of complexity. How can the growing set of functionality be used efficiently, with few errors, and to the users' satisfaction? Multi-touch and direct manipulation have already simplified and at the same time enriched the user experience, compared to formerly used menu navigation or stylus-operated desktop analogies, which were not adapted to mobile conditions. Yet, there is room for further improvement. In the manifold and fundamentally different use cases for mobile interaction (some of which emerged just recently), touchscreen input is not always the best option. Furthermore, the heterogeneity of users (from young to old, possibly impaired, etc.) must be addressed.

One approach to leverage mobile interaction (besides evident measures such as the application of usability rules [308], platform guidelines[2], etc.), is using *multimodality* (see Section 2.1.1 for the definition). This approach will be investigated in depth in this dissertation. Multimodal user interfaces can include, e.g., touch, motion gestures, speech, sound, alternately or in parallel [255]. Multimodal interaction has proved to have many advantages (see Section 2.2.1), but was hitherto mostly researched and used in desktop computing. However, particularly state-of-the-art smartphones and tablets are predestined for the use of rich multiple interaction modalities. With the multitude of built-in sensors, a vast design space of explicit and implicit sensor-driven interaction methods is opened up.

Recently, novel modalities are beginning to be employed in the mobile area. For example, users can scroll content with their eyes[3] or shake the device to undo actions[4]. Such task-

[1] http://www.apple.com/iphone/, last accessed July 31, 2014

[2] iOS Human Interface Guidelines. https://developer.apple.com/library/ios/documentation/userexperience/conceptual/MobileHIG/MobileHIG.pdf, accessed May 6, 2014

[3] Samsung Smart Scroll. http://www.samsung.com/us/support/SupportOwnersFAQPopup.do?faq_id=FAQ00053081&fm_seq=62452, accessed April 14, 2014

[4] Apple, Undo and Redo. https://developer.apple.com/library/ios/documentation/userexperience/conceptual/mobilehig/UndoRedo.html, accessed April 14, 2014

specific interaction modalities are however rarely used, as we show in a user survey conducted in this thesis (see Section 6.2.1). Furthermore, a survey of the 100 first-ranked Android applications in Google's Play Store revealed that the employment of sensory interaction remains sparse [92].

To this end, the underlying motivation of this work is to promote the use of multimodality in mobile interaction, in order to achieve a better overall user experience. We argue that multimodality can both improve interaction for existing use cases, and facilitate entirely new use cases and applications. The dissertation's contributions are twofold: making multimodal interaction more usable (from a user's perspective) and more accessible (from a developer's perspective).

In this work, we extend the term of multimodal interaction beyond the notion of modalities as interaction channels. In our considerations, we also include sensor-driven interaction, covering both explicit and implicit interaction [300]. Making use of sensors can capture information actively (e.g., by moving or tilting the device, as we will show in Chapters 3 and 5) or passively (e.g., through implicit context information, as we will show in Chapters 4 and 5), in order to interact with the device and with the environment (or objects therein).

1.2 Research Questions

We formulate the following high-level research questions (HRQ) that shall delineate the scope of this dissertation:

HRQ1 How can mobile interaction benefit from multimodality and sensor-driven interaction?

HRQ2 What are potential problems and challenges of multimodality and sensor-driven interaction?

HRQ3 How can applications for particular use cases be improved through multimodal interfaces?

HRQ4 How can users be better supported in making use of multiple modalities on their devices?

HRQ5 How can the implementation of multimodal and sensor-driven applications and interaction methods be better supported?

HRQ6 Which guidelines and lessons learned can be extracted for the design, implementation, and evaluation of multimodal and sensor-driven applications?

With the help of the above research questions, we approach our overall goal of leveraging mobile multimodal interaction in three dimensions: the *perspective, abstraction* and *interaction* dimension (visualized in Figure 1.1).

The *perspective* dimension represents two opposite points of view: the user's and the developer's perspective. On the one hand, we are interested in HCI-related aspects of multimodal user interfaces (UIs), such as efficiency, effectiveness, and satisfaction (user perspective).

On the other hand, we aim at supporting the development process (design, implementation, evaluation) of multimodal interfaces on a tool layer (developer perspective).

The *abstraction* dimension describes the scope in which we investigate multimodal interaction. We are interested in evaluating individual interaction methods or *modalities* in specific application domains to gain a deep understanding of their particular benefits and specialties for individual use cases (specific level). We though look also at general (system-wide) modality usage, investigate real-world use, and propose user interfaces to define and get noticed about multimodal behavior (general level).

Finally, the *interaction* dimension implies that we consider both multimodal *input* methods and interaction paradigms, as well as *output* modalities by which the system communicates with the user.

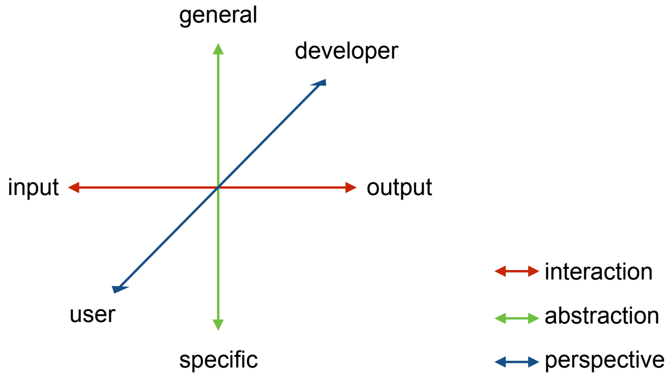


Figure 1.1: Dimensions in which this thesis tackles the problem of usable multimodal interaction

The high-level research questions will be investigated in the individual chapters of the thesis. At the beginning of each chapter, the main problem statement and subordinate research question that will be studied in this chapter is introduced. At the end of each chapter, we summarize the answers to the formulated problems and discuss the lessons learned.

1.3 Main Contributions

This dissertation makes the following novel contributions in the field of mobile multimodal interaction:

- We gain a deeper understanding of multimodality and its benefits in different application areas. At the same time, we reveal novel ways how multimodal interfaces can be used in these domains, and confirm their utility based on evaluations and user studies.
- By having chosen representative application areas, we show the design space of multimodal and sensor-driven interfaces in different social settings (private, semi-public and public environments).
- We conceived and implemented a rule-based model for the realization of multimodal behavior, supporting input as well as output, in everyday and specialized use cases.

- We created a multimodality programming framework which is based on the above model, supporting developers to implement multimodal interaction methods and realizing system-wide multimodal behavior.
- We propose and evaluate user interfaces for defining multimodal behavior and for creating awareness on the multimodal state of a mobile device.
- We provide recommendations and strategies for evaluating multimodal systems based on our research experiences.

Relation to the Software Development Life Cycle

While there exist plenty of software development process and lifecycle models (for a generalized view [141] or for individual classes of systems, e.g., perceptive user interfaces [163]), they can be abstracted to the most important phases *analysis*, *design*, *implementation*, and *evaluation* (see Figure 1.2).

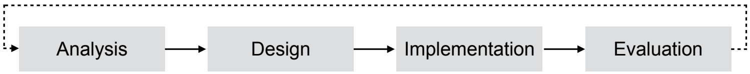


Figure 1.2: Archetypical model of the software development process. For multimodal systems, all parts are covered by this dissertation.

The tools presented and findings gained in this dissertation address all of these phases for multimodal and sensor-driven interaction. We demonstrate the utilization of various methods for the *analysis* (or planning/requirements elicitation) phase, like focus groups or large-scale questionnaires. We extensively report on the *design* (or conception/prototyping) phase of multimodal systems, often involving multiple stages. We thereby underline the iterative character of software and systems development. The *implementation* phase is not only addressed by the description of various individual systems, but also by the formalization through the above-mentioned multimodality programming framework. Finally, covering the *evaluation* phase, our findings are substantiated by user studies (laboratory experiments and real-world studies, short task assignments and long-term evaluations).

1.4 Thesis Outline

Figure 1.3 gives a graphical overview of the relationships between the chapters of this work. The dissertation is structured as follows.

In Chapter 1 (the present chapter), we motivate the topic of the dissertation and introduce the high-level research questions we will answer in the course of the thesis. We also briefly list the main contributions this thesis makes.

Chapter 2 provides the necessary background for this work. This includes, in the first place, the introduction and definition of basic terms that will be important in the remainder of the thesis. Furthermore, this chapter serves for situating this dissertation in the body of related work. We report on the state of the art in the areas that are related to this work: the sensors and actuators available in mobile devices (which lay the ground for multimodal input and output), the present tool support for modeling and implementing multimodal systems,

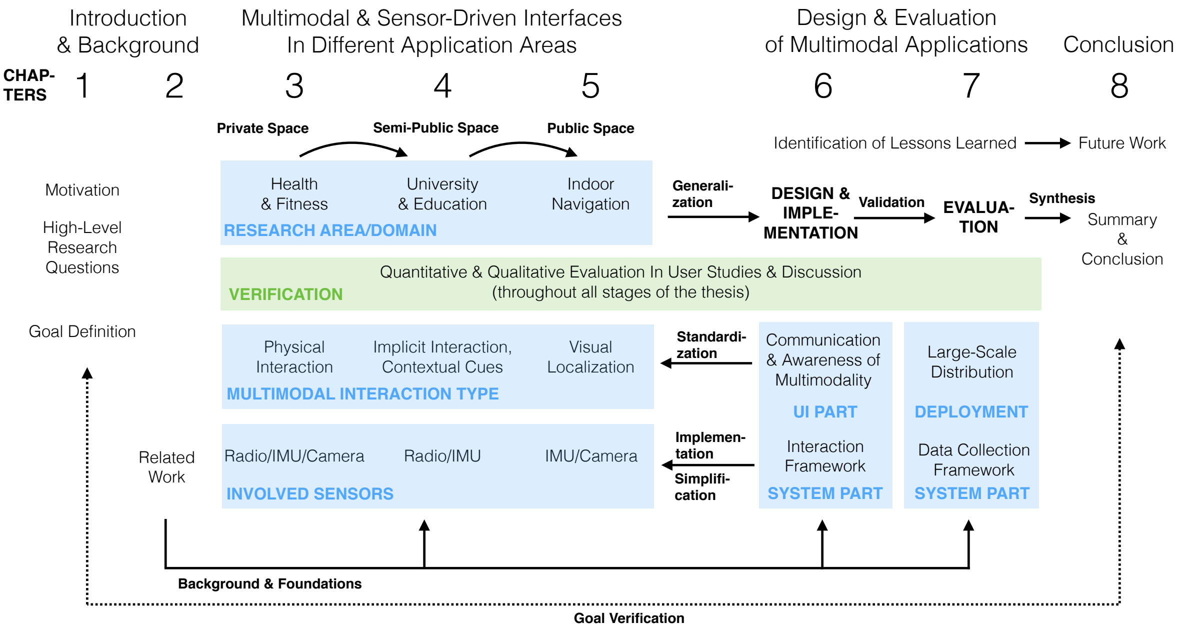


Figure 1.3: Visual structure of the dissertation. This diagram shows on a high level what will be subject of the individual chapters of this work, and how the parts are interconnected.

and the candidates for evaluation methods available and applicable for multimodal systems. Furthermore, we report on current systems in the different application domains that will be described in Chapters 3–5. That way, the reader will gain an overview on how problems in the individual application areas are addressed as of now, before we introduce our proposed approaches in the respective following chapters. The background chapter is organized such that its subsections accumulate the related work for all following sections in the thesis.

In the following three chapters (3, 4, and 5), we delve in depth into three distinct application domains (health & fitness, university & education, and indoor navigation). We chose them as representative areas of research, each of them currently being in the focus of public or scientific interest, so that they illustrate the design space of potential use cases for multimodal interaction. For each domain, we present work that comprises novel multimodal and sensor-driven user interfaces, addressing the particular requirements of this area. The presented approaches are at the same time representative examples of multimodal interaction in three different scopes (public, semi-public and private space). The approaches introduced in this thesis involve various sensors, on top of which different interaction methods and paradigms are implemented. We exemplify the applicability and benefits of multimodal interaction for these domains, as we have evaluated all approaches in online, laboratory, or real-world studies. At the end of each of these chapters, we distill the lessons learned and recapitulate the most important points with regard to the research questions formulated at the beginning of the chapter.

After having studied multimodality in individual application domains, a generalization and abstraction follows in the two consecutive chapters (6 and 7). They deal with two important steps in the development process of multimodal systems: design and evaluation. Chapter 6 (design) first analyzes problems with current multimodal systems, both from a developer's and an end user's perspective, and offers solutions for each of them. On the developer side, a programming framework is introduced, simplifying the integration of different input and output modalities. We validate the utility of the framework and discuss how the implementations presented in Chapters 3, 4 and 5 could be eased and extended with it. On the end user side, we propose and evaluate user interfaces for defining and modifying multimodal behavior, and for creating awareness on modality switches.

Chapter 7 is dedicated to the evaluation process. We first discuss to what extent evaluation of multimodal systems differs from conventional systems and portray selected evaluation techniques that have proven useful in the course of our research, which are *Wizard of Oz* (WOz), *Logging* and *Self-Reporting*. We report on our experiences with these techniques and delve deeper into two particular important questions by dedicated user studies: the accuracy of self-reporting in long-term studies, and the usage of app stores for large-scale deployment of research applications.

In Chapter 8, we conclude the dissertation by assembling the contributions of the individual chapters, referring to the initial high-level research questions. We also spot aspects that are beyond the scope of our work and were not treated in this thesis, and outline directions for future work.

Chapter 2

Background and Related Work

2.1 Foundations

As we investigate multimodal and sensor-driven user interfaces in this dissertation, we need to introduce in the first place the necessary terms and the background.

2.1.1 Terms and Definitions

We begin with giving definitions for the most important terms we will use in the remainder of this dissertation, including *multimodality* and *multimodal user interfaces*.

The term *modality* has different notions depending on the (psychological, medical, linguistic, etc.) perspective. In a very common sense, it denotes "the type of communication channel used to convey or acquire information" according to Nigay and Coutaz [246].

For our research in the context of human-computer interaction (HCI), two notions are especially important. They are related to input and output in the communication between humans and technological devices. First, modality can refer to the *sense* through which a user perceives the output of a system. These are the human senses vision, audition, touch, taste and smell (and further proprioception, thermoception, nociception and equilibrioception). Second, it can refer to the channel through which a system receives input from a user. Here *technological* input methods or input devices are meant. As Jaimes *et al.* [143] note, some of them correspond to human senses (e.g., a camera to the sense of sight, a microphone to the sense of hearing, a touch sensor to the sense of touch), but for others no direct mapping is possible (e.g., touchscreen, keyboard, hardware button, mouse, motion gestures). In this work, we subsume both meanings in the term *modality* and use the following definition:

Definition 1 (Modality) *Modality denotes a communication channel (in the sense of a human sense or an interaction method) between the user and a technological device.*

Similarly, there exist different definitions for multimodal systems. According to Bourget [33], "multimodal interaction refers to interaction with the virtual and physical environment through natural modes of communication such as speech, body gestures, handwriting, graphics or gaze." This definition inherently comprises the goal that multimodality shall make interaction natural, and thus user-friendly. Lew *et al.* [186] are less constrictive, saying that a multimodal HCI system is "one that responds to inputs in more than one modality or communication channel". They, however, disregard output modalities in their definition.

Coutaz [62] (extended in [246]) classifies multimodal systems along the dimensions *fusion* and *time*. In terms of fusion, multimodal systems can be independent (alternative modalities can be chosen to accomplish a task) or combined (multiple modalities are needed to accomplish a task). In the temporal dimension, systems can be sequential (multiple modalities are used one after another) or parallel (multiple modalities are used concurrently or synergistically). Refer to Section 2.1.3 for an (exemplary, yet not complete) overview of multimodal user interfaces.

For our research, we formulate the following comprehensive definition of multimodal interaction, where the term *modality* is meant to be understood in both notions of Definition 1 (human sense and interaction method).

Definition 2 (Multimodal Interaction) *Multimodal interaction denotes interaction with a system involving more than one modality for input, output, or both. The modalities can be used independently or combined, in parallel or sequentially.*

Especially in mobile devices, input and output modalities are implemented through or directed by sensors. This explicitly includes hardware sensors measuring a physical quantity and "virtual" software sensors, which, e.g., determine location or certain contextual data. (An overview of sensors integrated in state-of-the-art smartphones is given in Section 2.1.2.) Let us therefore introduce the term of *sensor-driven interaction*.

Definition 3 (Sensor-Driven Interaction) *Sensor-driven interaction denotes the communication with a system initiated or mediated by information acquired from sensors.*

The relationship between *multimodal interaction* and *sensor-driven interaction* is as follows. An input modality can be entirely covered by one sensor (e.g., by the camera for the visual modality), but it is also possible that fused information of several sensors at a time forms one new modality (in the sense of an interaction technique or paradigm [121]). For example, a motion gesture could be detected by fused accelerometer and magnetometer readings.

Sensor-driven interaction is situated on a continuum between *explicit* and *implicit* interaction. An example for explicit interaction is motion input, such as shaking the device. Implicit interaction denotes the situation where an action is performed based on context information that is not directly influenced by the user, such as location or ambient light level. An interface that adapts to the user's walking speed would be an example lying in between. In that case, the user performs an action (walking) that has a consequence on the user interface, but does not explicitly interact with the device. Such behavior is extensively described in the context *ambient intelligence* or *proactive computing* (see, e.g., [291] for a review).

Output modalities can likewise be sensor-driven, e.g., in a way that contextual cues determine information presentation. For example, the device could switch between visual and auditive notifications depending on the ambient noise level. In this case, implicit information often plays an important role, too.

To best describe the focus of our research, we want to coin a term for a user interface that 1) is multimodal in the above described sense and 2) additionally includes the following aspects that are not implied in existing definitions of multimodality:

- that multimodality is (partly or entirely) facilitated by device-internal sensors, i.e., that data from these sensors are used to implement the modalities

- that modalities are understood as (sensor-driven) interaction techniques
- that the user interaction can be (partly) implicit, or that the explicit user interface is influenced by implicit information.

We call this new class of interface a MUSED (MUltimodal and SEnsor-Driven) user interface and define it as follows:

Definition 4 (MUSED User Interface) *A MUSED user interface allows the user to communicate with a system by multimodal and/or sensor-driven interaction in the sense of Definition 2 and Definition 3.*

The abbreviations MUSED user interface or MUSED interaction will be used in the remainder of this thesis when one of the above listed aspects shall be emphasized, or when we clarify that the described interface goes beyond the traditional notion of multimodal interaction.

2.1.2 Sensors and Actuators in Mobile Devices

In order to help define a design space for MUSED interaction, we give an overview of sensors and actuators that are available in current smartphones[5]. In this overview, we include technologies and standards that are currently prevalent in commercial devices and/or supported by mobile operating systems[6,7,8]. We also consider selected technologies still being in the research phase, but do not list sensors or standards that are mostly used in other domains (e.g., home automation).

Sensors

Sensors are, as outlined in the previous section, the basis for the implementation of mobile device input modalities. Lara and Labrador [176] distinguish four attributes that can be sensed in the context of activity recognition with wearable sensors: environmental attributes, acceleration, location, and physiological signals. For smartphone sensing, we adapt their classification slightly and distinguish the following five categories: contact sensors, motion sensors, environmental sensors, position sensors, and radio communication. Besides an overview both on hardware sensors (that measure a physical quantity, such as acceleration), we also address virtual sensors (that return contextual data provided by an aggregation of hardware sensors or by other information sources).

Contact Sensors

Contact sensors measure a physical contact of the user's hand, finger, or other body parts with the device, which is the case for, e.g., touchscreen interaction or the fingerprint sensor.

[5] This information refers to the state of the art in 2014, when this dissertation was written.

[6] `http://developer.android.com/guide/topics/sensors/sensors_overview.html`, accessed February 21, 2014

[7] `http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh202968(v=vs.105).aspx`, accessed May 8, 2014

[8] `https://developer.apple.com/technologies/ios/features.html`, accessed July 31, 2014

- **Touchscreen**: Being a combined in- and output interface, the touchscreen is usually the prevalent way for interacting with a smartphone. Today's smartphones usually employ capacitive touchscreens, while older devices used resistive touchscreens (working via pressure).
- **Fingerprint Sensor**: Recently, some manufacturers have started to include a fingerprint sensor that not only allows unlocking the phone without entering a passcode, but also offers shortcut actions like confirming app store purchases (e.g., Apple iPhone 5s) or launching predefined apps with different fingers (e.g., HTC One Max, Samsung Galaxy S5). Depending on the sensor type, a fingerprint is read by touching (Apple) or swiping the finger over the sensor (HTC, Samsung). If the sensor is made accessible to developers via according APIs (Application Programming Interfaces), one could imagine manifold use cases beyond authentication.
- **Physical Controls**: Many devices have mechanical buttons or switches for turning the device on and off, controlling the volume, muting and un-muting the device, or performing shortcut actions (e.g., starting applications or returning to the home screen). Through the provided haptic feedback, they can be operated without looking at the screen.

Motion Sensors

Most of today's mobile phones have motion or inertial sensors. While a basic use case for motion sensors is detecting the rotation between portrait and landscape mode and accordingly adapting the screen orientation, these sensors support a wide range of interactions based on device motion, including activity recognition [176]. The combination of accelerometer and gyroscope often is referred to as inertial measurement unit (IMU).

- **Accelerometer**: This sensor measures the linear acceleration force that is applied on the device in three dimensions. Therefore, devices usually have a 3-axis accelerometer. The unit of these sensor readings is *meters/second*^{2}. The acceleration forces can be used to detect motion of the device, such as shaking, tilting, etc. This also includes the force of gravity. APIs can provide a "software" gravity sensor (e.g., in Android) that explicitly provides the force of gravity. The translational motion can be calculated by double integration.
- **Gyroscope**: This sensor measures the Coriolis force due to the rate of rotation (angular velocity) around the x, y and z axes. The unit of these sensor readings is *rad/s*. The rotational speed can be calculated by integration. These forces can be used to detect rotation of the device, such as spinning or turning.

Environmental Sensors

In this category we list sensors that provide information about the physical environment.

- **RGB Camera**: As a mobile device's "eye to the world", the camera is one of the most important environmental sensors (and probably the most important sensor besides networking/radio communication). This can be said in light of the fact that the camera is incrementally used for more purposes than just taking photos. As one example, in Chapter 5, we use the camera as a basis for (visual) indoor localization. Two major camera principles are currently employed: Images are captured by a CCD (charge-coupled device) or CMOS sensor (active pixel sensor produced by a complementary metal-oxide semiconductor) containing an array of pixels. Many smartphones meanwhile have a front and a back facing camera, or even two back cameras to record stereo images or to allow re-focusing recorded images. Another way to acquire three-dimensional information are plenoptic cameras such as demonstrated by Lytro [112] or by Pelican Imaging [331]. Using an array of microlenses, directional information of the inciding rays of light (light field) can be captured. Such devices are expected to be included in smartphones in 2014[9], which will likely leverage computer vision and 3D reconstruction applications.

- **Infrared Camera**: Infrared or thermal imaging cameras can sense infrared radiation and became widely popular with the Microsoft Kinect[10], enabling 3D perception (in combination with an infrared projector) in a consumer price range. Google presented the integration of an infrared-based depth sensor in a smartphone in Project Tango[11], allowing three-dimensional perception of the environment. This could enable a plethora of novel applications and interaction possibilities, from gaming to indoor navigation. In a prototype by metaio[12], a thermal imaging camera visualizes the slight temperature difference of a spot on a surface that has just been touched. The system combining an infrared camera with AR (Augmented Reality) projection can thereby render any surface to a touchscreen.
- **Light sensor (Photometer)**: A photodiode measures the level of ambient light in *lux*. It is often used to auto-adjust the brightness of the screen, but can also be the basis for inferring other context information (e.g., activity).
- **Proximity Sensor**: This sensor detects when the device is close to another object. Often, the proximity sensor is used to switch off the screen when the user brings the smartphone in proximity of the ear. While the light sensor in combination with a threshold can be used as proximity sensor (illumination beyond a certain threshold will be interpreted as an object close to the sensor), it can also be realized using an infrared LED (light-emitting diode) and a photodiode that measures the level of reflection of infrared light [213]. Proximity can also be used for “pre-touch” interaction with the screen. Using a particularly sensitive capacitive touch screen, the device can distinguish whether a finger is closely above the display or actually touching it, which is called “Floating Touch” (Sony) or “AirView” (Samsung).
- **Magnetometer**: The magnetic field sensor measures magnetic fields in the unit *tesla*. By using a rotation matrix, the orientation of the device can be calculated, which realizes a *compass*.
- **Microphone**: The microphone records sound by transforming acoustic pressure in the air to electric signals. It can not only serve to realize a dedicated input modality (speech input), but can also sense the ambient sound level, which serves, e.g., for adapting

[9] http://www.engadget.com/2013/05/02/pelican-imaging-array-camera-coming-2014/, accessed May 30, 2014

[10] http://www.microsoft.com/en-us/kinectforwindows/, accessed May 30, 2014

[11] http://www.google.com/atap/projecttango/, accessed February 21, 2014

[12] http://www.metaio.de/press/press-release/2014/thermal-touch/, accessed May 30, 2014

output volume or modalities. Modern smartphones often have multiple microphones for improved filtering of environmental noise.

- **Barometer**: This sensor measures the air pressure in *bar*, which can also be used as the basis for calculating the elevation. A common use case is to measure altitude differences to detect floor levels. One of the first devices to be equipped with a barometer was the Samsung Galaxy Nexus.
- **Thermometer**: Some smartphones (e.g., the Samsung Galaxy S4) have a dedicated sensor to measure the ambient temperature. Many other devices have a built-in temperature sensor as well, but their purpose is to measure the temperature inside the device to protect hardware from overheating. However, apps like Temp Now[13] calibrate the thermometer with at least two reference temperature values so that the app can estimate the ambient temperature based on the internal thermometer readings.
- **Hygrometer**: This is a humidity sensor measuring the relative ambient humidity, built-in, e.g., to the Samsung Galaxy S4.
- **UV Sensor**: This sensor measures the ultraviolet (UV) radiation of the sunlight and is, e.g., included in the Samsung Galaxy Note 4 and the Samsung Gear S smartwatch. Samsung's S Health app provides precautions based on the measured UV level.
- **Laser Scanner**: Laser light is already used in commercial devices for measuring the distance to objects, which allows quicker autofocus when taking a photo[14]. In near future, laser scanners will be miniaturized and integrated in smartphones, so that phones can create a highly accurate 3D model of the environment using LIDAR (LIght Detection And Ranging)[15].

Radio Communication

Smartphones support a range of wireless communication standards which work using radio communication. We list the most important ones here, as they allow receiving information from the environment and can thus be considered as sensors (of radio waves) as well.

- **GNSS**: Global Navigation Satellite Systems (GNSS) receivers return the device's latitude and longitude with relation to a navigation satellite system, such as the US-American Global Positioning System (GPS), the Russian GLONASS (engl. translation: Global Satellite Navigation System), the Chinese BeiDou, or the European Galileo system. Using radio communication with GNSS satellites and trilateration, the position can be determined with few meters accuracy.
- **Mobile Telephony**: The most important standards for mobile communication are GSM (Groupe Spécial Mobile, later the acronym was changed to Global System for Mobile Communications), also referred to as 2G (second generation), UMTS (Universal Mobile Telecommunications System, 3G), and LTE (Long-Term Evolution, 4G). The packed-oriented data transmission standards in GSM are GPRS (General Packet Radio Service) and, with increased data rate, EDGE (Enhanced Data Rates for GSM Evolution). The

[13] http://www.imore.com/tag/temp-now, accessed May 6, 2014

[14] http://www.lg.com/de/handy/lg-G3, accessed May 30, 2014

[15] https://www.kickstarter.com/projects/ikegps/spike-laser-accurate-measurement-and-modelling-on, accessed May 30, 2014

data transmission speed in UMTS was enhanced by HSPA+ (High Speed Packet Access) [295]. This development coincides with the permanent need for more bandwidth through novel applications, such as, e.g., mobile video and audio streaming. However, the amount of data in recent technologies is growing faster than the available bandwidth (consider, e.g., state-of-the-art devices that have "Full HD" resolution displays and the according desire to watch "Full HD" content on these devices on the go). This is opposed to the fact that in 2014, 4G and even 3G coverage is by far not given everywhere (especially not in rural areas). Efficient bandwidth usage is therefore still important to consider in mobile application development.

- **WLAN**: Using Wireless LAN (Local Area Network), an IEEE standard (802.11), a mobile device can connect with wireless access points (infrastructure mode), as well as establish ad-hoc connections with other devices.
- **Bluetooth**: It is likewise an IEEE standard (802.15.1) for wireless short-range data exchange. Bluetooth profiles specify interfaces for different use cases, such as file transmission, internet connection sharing, or hands-free phone calls. With the introduction of the Bluetooth Low Energy (BLE) protocol stack[16], the power consumption can be significantly reduced.
- **DECT**: We list this standard for cordless home phones (Digital Enhanced Cordless Telecommunication) only for reasons of completeness as there are a few hybrid (Android) smartphones that support DECT as well. Besides telephony, DECT has been used, e.g., for localization [164].
- **RFID, NFC**: The RFID (Radio Frequency IDentification) system was created to identify objects and is suited for short-distance communication. Especially the NFC standard (Near Field Communication), which is based on RFID, is an established standard which is built into many smartphones and can be used, e.g., for mobile payment[17].
- **Infrared**: Widely used in the 1990s and early 2000s to exchange data between mobile phones or between handset and PC, infrared has become less important, being replaced by the significantly faster WLAN and Bluetooth. As of 2014, some devices begin to reintroduce infrared ports (e.g., the HTC One), supporting, e.g., remote control apps for consumer electronics. Infrared is also used to detect hand gestures (e.g., in the Samsung Galaxy S5) by the reflections of infrared rays from the user's palm.
- **ZigBee**: Personal area networks (PAN, as opposed to LAN) aim at short-range connections of wireless sensors, smart objects or devices in the home automation domain. One standard (IEEE 802.15.4) is ZigBee[18], which is planned to be included in Samsung and HTC smartphones.
- **ANT**: The ANT and ANT+ short-range standards have a higher data rate than ZigBee and are power-efficient at the same time. They are especially used to communicate with sensors from the health and fitness domain (e.g., heart rate or cadence sensors,

[16] marketed as Bluetooth Smart, http://www.bluetooth.com/Pages/Bluetooth-Smart.aspx, accessed February 24, 2014

[17] e.g., Mastercard Paypass, https://www.paypass.com/, accessed February 24, 2014, or FeliCa, a de-facto standard in Japan, http://www.sony.net/Products/felica/, accessed February 24, 2014

[18] http://www.zigbee.org, accessed February 24, 2014

blood pressure monitors, etc.), but also for home monitoring and automation. Several Android-based smartphones and tablets (e.g., Samsung Galaxy S4, Galaxy Note 10.1, Sony Xperia series) are equipped with built-in ANT+ communication.

Virtual Sensors

Under the term *virtual sensors* or *software sensors*, we understand data sources that provide context information or information on the environment including the user, although they are no real physical sensors. Often, they are accessible via APIs just like real sensors. An example is Android's orientation sensor[19], returning the x, y, and z orientation of the device in space. Although this sensor is accessible just as other hardware sensors by Android's `SensorManager`, its data is actually derived from the accelerometer and the magnetometer.

Similarly, the ability to determine the location can be regarded as a “location sensor”, although various hardware sensors and technologies can be used for that purpose (e.g., the GPS receiver for satellite-based localization, WLAN for access point triangulation and fingerprinting, or the camera for vision-based localization). A more extensive discussion about determining the location is provided in Section 2.2.4.

Actuators

Actuators can be used as a basis for output modalities, i.e., to communicate information to the user.

- **Screen**: As humans perceive about 80 percent of their environment by the sense of sight[20], the screen as *visual* output modality can be seen as the primary output device.
- **Projector**: Pico projectors likewise use the visual output channel, but extend the available display real estate as well as the social scope (a larger circle of users can participate) [114]. Smartphones with integrated pico projectors are commercially available, e.g., the Samsung Galaxy Beam[21].
- **LED(s)**: A singular LED, as available on many smartphones, can communicate information to the user, e.g., notify on incoming messages or missed calls without turning on the screen or making a sound. Some devices have multi-color LEDs, so that blinking patterns as well as color coding can be used to transport individual information. A characteristic aspect is that only the user knows the “matching” for which type of information a certain color or blinking pattern stands. Such a visualization is peripheral and privacy-preserving, similar to ambient user interfaces like the *Ambient Orb*[22].
- **Vibration Motor**: The usage of the *haptic* modality, enabled by the vibration motor, allows unobtrusive notifications without producing a sound or turning on the screen.

[19] `http://developer.android.com/guide/topics/sensors/sensors_position.html`, accessed May 6, 2014

[20] `http://www.brainline.org/content/2008/11/vision-our-dominant-sense_pageall.html`, accessed February 25, 2014

[21] `http://www.dlp.com/pico-projector/phone-projector/default.aspx`, accessed May 30, 2014

[22] `http://postscapes.com/ambient-orb`, presented in 1992 by Ambient Devices, Inc., `http://www.ambientdevices.com`, accessed February 25, 2014

Some mobile operating systems (e.g., Apple iOS) have the built-in possibility to define custom vibration patterns that can be assigned to different events.

- **Loudspeaker**: The loudspeaker uses the *auditory* modality to communicate via sound with the user.
- **Bone Conduction**: This technique transmits audible content to the inner ear through the bones of the skull. While Fukumoto [107] proposed to use the finger as transmission route to the ear, commercial solutions usually require a special headset. Bone conduction is, e.g., employed in Google Glass[23] and can also prove useful for hearing-impaired persons.

2.1.3 Mobile Multimodality

Input and Output Modalities on Mobile Devices

Modalities Involving One Sensor or Actuator

After we have given an overview of sensors and actuators available in state-of-the-art devices, let us look at some examples for input and output *modalities* in mobile user interfaces. Some of them can be considered as "common sense", some have been presented in a research context. In the latter case, we provide a reference to the respective publication. The following input modalities can be listed:

- Touch, performed on the device's touchscreen, to manipulate user interface elements (buttons, checkboxes, selections), or to type on a virtual keyboard
- Motion gestures, performed on the device's touchscreen (single- or multi-touch)
- Device gestures, performed with the device in space, such as shaking, tilting, or moving to a certain direction [165]. Device gestures are also referred to as extra-gestures, while motion gestures are called intra-gestures [339].
- Hand gestures, performed in front of the screen and detected by the Infrared sensor (e.g., in the Samsung Galaxy S5)
- Operation of hardware buttons and switches, e.g., to control the volume, take a photo, or enter text. Common hardware controls are volume buttons, mute switches, hardware keyboards, navigation keys, quick access keys, and trackballs.
- Pen input on the touchscreen, allowing handwriting, drawing, sometimes with Bluetooth connection for additional features like pressure sensitivity[24]
- Speech, for text input (dictation) or execution of commands by digital "personal assistants" (Siri, Google Now, Microsoft Cortana, Samsung S Voice, and similar)
- Physiological input, such as heart rate (e.g., in the Samsung Galaxy S5 or in smartwatches like the Samsung Galaxy Gear 2), brain interfaces [42], or skin input [118, 341]

[23] `http://www.google.com/glass/start/`, accessed May 30, 2014

[24] `http://www.adonit.net/jot/touch/`, accessed May 8, 2014

- Implicit and context-based input, such as location, time, weather, social setting, etc.

On output side, there exist the following modalities:

- Screen, as the primary modality using the visual channel
- Notification lights, variable in color and blinking pattern
- Sound, for audio cues (e.g., as feedback on performed action like a typing sound or a confirmative sound when a message was sent), as well as for music and speech output (e.g., consuming text using a text-to-speech engine)
- Haptic feedback [196], provided by the vibration motor

Modalities Involving Multiple Sensors or Actuators

Research has demonstrated in various ways how novel interaction modalities can be created from the combination of multiple sensors or actuators. We here only describe a few of them exemplarily. One of the first descriptions of multimodal task completion, although at that time not yet for mobile interaction, is the "Put that there" approach by Bolt [29]. The user gives a speech command and uses a gesture to select an object the command should be applied to. Wasinger *et al.* [339] showed a shopping assistant system transferring this to mobile systems. In a store, users can, e.g., point at an object and ask "How much does *this* cost?".

Touch Projector [30] is an interaction technique using both motion and device gestures. By pointing at an external display with a smartphone, the user can project touch interaction performed on the mobile device onto the distant screen. That way, the user can modify content on an (otherwise unreachable) wall projection, laptop, or public display, as if she had directly interacted with the distant screen.

With Sensor Synaesthesia [129], touch and motion are combined within a device in two ways. Touch-enhanced motion enables interaction techniques like "tilt to zoom", where the user touches the display and then tilts the device to zoom. On the other hand, motion-enhanced touch can detect the tap intensity of touch input by incorporating the accelerometer.

Harrison and Hudson [117] presented Scratch Input, an acoustic-based input method that detects gestures on various surfaces by the unique sound of scratching.

How to Choose Modalities?

Having shortly presented the design space for multimodal interaction, the next obvious, but non-trivial question is now to choose the right modalities for a certain situation or use case. Criteria for modality selection are related to their individual properties, as suggested by Ratzka [270, 271], based on Bernsen [24]. These criteria, which we will detail in the following, are: required *interaction channels*, *salience*, *local selectivity*, *control*, *learning requirements*, and *expressiveness* [270]. A first clue is that the information to be transported often is already associated with a certain channel (e.g., an illustration can only be perceived by the visual channel, while text can be perceived visually or acoustically). Another factor is *salience* [347]: For example, information on the auditive channel attracts attention more effectively than the visual channel. This effect is related to the *local selectivity* property: In order to perceive a visual notification, attention needs to be more directed than for an aural notification. In the visual domain itself, there are saliency effects [136] and the user's attention is directed, e.g., by Gestalt principles [46]. The desired amount of *control* affects the choice of a modality as well. The user can control reading speed for written text, whereas for speech output, the pace is determined by the system. Novel speed reading output methods such as Spritz[25] stream text and help to control eye saccades by highlighting the "optimal recognition point" of words. Such techniques reduce the level of control also for the visual modality. Furthermore, the *learning requirements* may vary for modalities. A modality may be easier to use if it is based on familiar concepts or real-world metaphors. Take for example the "pinch to zoom" gesture introduced with the iPhone: This gesture gives the user the impression to physically stretch an object to enlarge it, which was perceived as highly intuitive. As a comparison, the manipulation of three-dimensional objects in desktop modeling software with traditional input devices (keyboard, mouse) does not feel natural and requires training for most people. Finally, *expressiveness* is mentioned as criterion of how well a modality (or a combination thereof) communicates the information to be transported [24, 271].

Instead of looking at modality properties, Reeves *et al.* [274] proposed a set of goal-oriented guidelines when choosing the modalities for a system.

- The interface should be flexible enough to support the broadest range of users and contexts possible. This refers to the choice of alternative modalities. Users then can select the most appropriate mode of interaction depending on the situation (e.g., speech input when the hands are involved in another task, and touch input when ambient noise is high).

- Multimodal systems should be adaptive with relation to context, users and application needs. Such adaption could happen proactively or based on previously defined rules, as we will present in our multimodal interaction framework presented in Section 6.3.

- The general capabilities of users, as well as individual needs and preferences (such as disabilities), should be respected. This aspect relates especially to the combination of multiple channels to process information, which affects the efficiency of perception. Taking care of personal preferences may also lead to positive emotions when using the system, which contributes to a better user experience, according to Norman's findings on emotion and design [249].

- The multimodality of a system should contribute to error prevention on different levels. Concurrent modalities can increase the robustness, when, e.g., an overheard notification is additionally displayed visually. Further, the choice of modalities can reduce errors, as users may choose the most familiar, comfortable or reliable modality depending on the individual context.

- Privacy and security issues should be considered as well when choosing modalities. Security can be affected, e.g., by overhearing or shoulder-surfing. As an example, the use of speech input in public spaces could expose privacy-sensitive information and thus be undesired.

Lemmelä *et al.* [184] chose the approach to investigate the aural, visual, physical, cognitive and social load in different situations. Based on field observations, they identified suitable

[25] `http://www.spritzinc.com/the-science/`, accessed March 14, 2014

Preferred Input

		Speech input			Typing		
		Visually impaired people	Motor impaired people	Bad lighting conditions	People with speech disorders	Noisy environments	Public environments
Discouraged Input	Speech input						
	Hearing impaired people	X	X	X			
	Noisy environments	X	X	X			
	Public environments	X	X	X			
	Typing						
	Visually impaired people				X	X	X
	Car user interfaces				X	X	X
	Glaring environments				X	X	X

(a) Input constraints

Preferred Output

		Speech output			Visual text		
		Visually impaired people	Car user interfaces	Bad lighting conditions	Hearing impaired people	Noisy environments	Public environments
Discouraged Output	Speech output						
	Hearing impaired people	X	X	X			
	Noisy environments	X	X	X			
	Public environments	X	X	X			
	Visual text						
	Visually impaired people				X	X	X
	Car user interfaces				X	X	X
	Bad lighting conditions				X	X	X

(b) Output constraints

Figure 2.1: Constraints for input and output modalities (exemplary for speech and typing input, and speech and visual text output) based on contextual and user-specific factors. The crossed out sections mark conflicting usability goals. Own visualization based on [270].

contexts for selected output modalities. Ratzka [270] suggests that the choice of modalities should be informed by the concrete tasks an application is used for, which can be understood as a concretization of universal guidelines as above. Ratzka further argues that input and output modalities underlie various constraints, which Figure 2.1 visualizes in an exemplary manner [270]. The matrices show possible combinations of modalities in different situations, where inappropriate combinations are crossed out. The designer can use the matrices, according to Ratzka, to check "whether for each individual candidate modality the factors listed in the columns outweigh the factors listed in the rows and contrast these results for each interaction modality" [270]. This method, however, does not provide recommendations how to combine modalities.

Besides the already discussed approaches of applying guidelines and principles [24, 184] and analyzing the task [270], user interface patterns are a further way to find appropriate and efficient combinations of modalities [270, 271]. Design patterns are an established technique often used in software engineering, and have the advantage of being proved in practice as they have successfully been applied in prior work [270].

In our work, we conducted a comprehensive survey on the usage of both input and output modalities. We report on the results in Section 6.2.1. Since our findings outline the heterogeneity of users and of their modality preferences, we argue for a solution that leaves flexibility to the users by their own definition of multimodal behavior. Our respective solution is presented in Section 6.4.

User Acceptance

One additional factor that may not be neglected is the users' attitude towards multimodal interaction. Prior research has often shown that users tend to stick with familiar approaches or modes of interaction, rather than adopting something new [221]. This effect can be particularly strong for entirely new modalities. Evaluating the usability of completely novel approaches is more difficult than of incremental improvements [131]. Hence, the benefit of novel interaction modalities must become clear to users in order to achieve wide acceptance.

Ruiz *et al.* [289] conducted a study in which subjects should propose device gestures for performing certain actions. Subjects preferred natural use and real-world metaphors, e.g., bringing the phone to their ear for the action "answer a call" or place the phone with the screen facing down to "end a call". The study suggested that especially the usage of motion gestures in general found broad acceptance. Only 4% of subjects stated that they would never use motion gestures.

A further factor for acceptance is social acceptability. Rico and Brewster [275] evaluated the social acceptability of different motion gestures and found that input methods "mimicking gestures encountered in everyday life" [275] are more likely to be accepted.

2.2 Exemplary Application Domains for Mobile Multimodal Interaction

We will now address three individual application domains (health & fitness, university & education, and indoor navigation) that are potential candidates for MUSED interaction, and report on the state of the art in research and industry in these domains. This section is partly based on related work overviews provided in our prior publications. Each of the subsequent chapters (Chapter 3, 4, and 5) will then be dedicated to one of these areas, and we will present multimodal approaches addressing individual problems in these domains.

Before that, we will briefly motivate the use of multimodal interaction from a general point of view.

2.2.1 Why Using Multimodal Interaction?

There are a number of reasons why multimodal interaction can be beneficial. The reasons we list here do not address individual application areas, but mark advantages from a general point of view.

Naturalness

Humans communicate using all senses, so that their interaction with the world is inherently multimodal [38, 266]. The choice of communication modalities is however partly unconscious, so that an interface seems more natural when multiple modalities are supported [340]. Thus, multimodal interaction also adds to more intuitiveness [146].

Efficiency

Given that in multimodal interaction, different modalities can be used simultaneously [246], a higher bandwidth of information transmission is available. This makes multimodal interaction in principle more efficient than unimodal interaction, as more information can be communicated in the same amount of time. Particularly in mobile settings, individual unimodal interaction modes can suffer from limitations (e.g., the "small screen" problem when relying only on the visual channel). In this case, multimodality can have a synergetic effect and contribute to more efficiency [255]. The efficiency boost, however, depends on the concrete use case, as other evaluations [254] showed only small increases in task completion time.

Robustness

A multimodal user interface is expected to be more robust to errors. The reason is that the risk for perceiving a piece of information erroneously or missing information is minimized when the same information is communicated redundantly over another modality. This is, e.g., the case when notifications are sent both by a visual alert and a sound. Likewise, robustness can be increased if the user can choose out of several input methods the least error-prone in a particular situation. Oviatt [254] showed that users made about one third fewer errors with a multimodal interface than with a unimodal interface. This stands in contrast to the fact that in actual systems, information is to the most extent transferred using one modality, namely the visual channel [248].

Adaptivity to Information

It depends on the type of information by which modality it is transported best [266]. User interfaces that provide multiple modalities thus are likely to better communicate a broad range of information types than unimodal interfaces.

Adaptivity to Cognitive Resources

Chittaro [50] notes that "physical parameters (illumination, noise, temperature and humidity, vibration and motion, ...) of the mobile user's environment are extremely variable, limiting or excluding one or more modalities. 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 colors on the screen or even to read the screen at all, on a moving vehicle we might not notice vibrations generated by the device". In mobile settings, interacting with the smartphone is not a primary task, as users will have to focus their attention on the environment (traffic [77, 78], social interaction [319], etc.). Their cognitive resources are thus limited [50]. Redundant multimodal interfaces can address these limitations, as users can choose an optimal modality according to the physical parameters and the available resources [184].

Diversity

Multimodal interfaces can improve interaction not only when subjects have cognitive, but also physical limitations. This includes, e.g., motor impairments (e.g., reduced touching accuracy or fine motor skills), or vision problems (e.g., reduced ability to read small text or to distinguish colors – 8% of all males suffer from red-green blindness[26]). In this case, alternative interaction modalities (e.g., speech) can be used. Motor limitations can also be caused by mobile settings, e.g., "accelerations and decelerations of a vehicle subject passengers to involuntary movements, which interfere with motor operation of the device (e.g., time and errors in selecting options, effectiveness of using writing recognition or gesture recognition software, etc.)" [50].

Taking these aspects into account, multimodal user interfaces contribute to diversity-friendly design, respecting requirements of heterogeneous user groups and heterogenous usage contexts.

[26] http://www.color-blindness.com/2010/03/16/red-green-color-blindness/, accessed May 9, 2014

Popularity

Although a multimodal user interface does not necessarily imply that users will interact multimodally [255], multimodal systems are often very popular with users. For example, Oviatt [254] found that 90% to 100% of users prefer multimodal interaction over unimodal interaction. Wolff *et al.* [347] similarly found a high preference for multimodal interaction. They let subjects choose between unimodal and multimodal commands in the visual/spatial domain and found that only three out of 98 were performed unimodally. Even if systems do not require the use of multiple (parallel) modalities, but offer modalities as alternatives, they "empower the users", giving them a feeling of control. This "internal locus of control" is, according to Shneiderman [308], one of the crucial factors to satisfaction with a user interface. However, Lemmelä *et al.* [184] underline that discoverability is an important criterion for novel modalities: Users must be able to recognize which interaction methods are available, and how they have to be used.

Social Acceptance

The social contexts or social norms influence which kind of interaction with a mobile device is appropriate. Chittaro states: "[K]eeping sound on at a conference is not tolerated, while looking at the device screen is accepted; keeping a lit display in the darkness of a movie theatre is not tolerated; making wide gestures while being near strangers might be embarrassing in general and not advisable in specific places" [50]. A limited choice of modalities might disqualify the use of devices or applications in certain social settings. Multimodal interaction can solve this problem by offering interaction forms that are socially acceptable [275].

Novel Possibilities

Besides improving interaction of existing applications, multimodality also enables completely novel interaction paradigms, applications, and functionality [89].

We conclude with a quote by Dumas *et al.*, who summarize multimodal systems as follows: "[They] represent a new class of user-machine interfaces [that] tend to emphasize the use of richer and more natural ways of communication, such as speech or gestures, and more generally all the five senses. Hence, the objective of multimodal interfaces is twofold: (1) to support and accommodate users' perceptual and communicative capabilities; and (2) to integrate computational skills of computers in the real world, by offering more natural ways of interaction to humans." [89]

In the following, we illustrate the state of the art in mobile interaction in selected example domains (their choice has been motivated in Section 1.4). The subsequent chapters (3, 4 and 5) will then investigate the room for improvement with the help of multimodal and sensor-driven interaction.

2.2.2 Health, Fitness, and Daily Activities

In the recent past, one could observe a growing interest in health and fitness support by mobile devices. This effect has to do with two factors. First, there is a trend to monitoring health- and fitness-related data, as part of the "Quantified Self" movement[27]. This is reflected by an increase of sensor-equipped mobile devices and accessories for that purpose, such as run trackers, glucose meters, or heart rate monitors. The latter is even directly built into smartphones like the Samsung Galaxy S5. Second, it is a consequence of the aging society that more and more people suffer from multimorbidity [327], i.e., two or more diseases at a time, and need temporary or permanent support in their activities of daily life. Mobile devices can support caregivers or patients in many cases to, e.g., allow elderly people to continue living autonomously in their homes. This research area is also referred to as ambient assisted living (AAL) and computer-supported coordinative care (CSCC) [58].

Health and Ambient Assisted Living

We can distinguish between systems addressing caregivers and systems to support people directly in their daily lives. For caregivers, Mynatt *et al.* [240] proposed the CareNet Display, a digitally enhanced picture frame with an image of the loved one. Icons placed around the frame show caregivers whether their loved ones have taken their medications, eaten their meals, performed physical activities, etc.

Systems can also directly support (not only, but including elderly) people in the so-called activities of daily living (ADL). In the food domain, calorie monitoring apps (e.g., by food ingredient databases like FDDB[28]) help pursuing a healthy diet; and sensor-enhanced kitchen knives and cutting boards [172] can even recognize types of vegetable being cut, and thereby estimate which meals have been prepared. Home appliances can be automated or remote-controlled. For example, in a smart home, smartphones can turn on and off the lights, helping especially elderly people who have problems with walking or climbing stairs. Systems that monitor the remaining time until the laundry in the washing machine or the meal in the oven is finished [199] additionally shorten ways in the house, as residents do not have to check multiple times by themselves if the laundry or the meal is done. A methodology and toolchain to prototype mobile interaction within intelligent environments has been investigated by Diewald *et al.* [84].

There exist also smartphone apps that address specific medical conditions. Fontecha *et al.* showed a system to detect frailty [102] based on accelerometer readings. Armstrong *et al.* provide use memory cues in smartphone applications for Alzheimer's disease patients [10] to stimulate autobiographical memory, an idea which is also pursued by life logging with a wearable camera [130]. Du *et al.* present an Android-based healthcare management system that calls support in emergency cases [87]. Bardram *et al.* created an app-based monitoring system for patients with bipolar disorder [17]. Smartphone medication adherence apps [71] address the problem of decreasing memory. They remind patients when to take in their pills respecting their daily routines, or provide real-time feedback for medication intake [180, 312].

While the research exemplified above usually focuses on functionality, Gao and Koronios [109] point out that when developing for elderly people, special requirements for usability

[27] http://venturebeat.com/2012/01/18/the-view-from-ces-the-top-trends-in-technology-for-2012/view-all/, accessed August 19, 2014

[28] http://fddb.info, accessed May 31, 2014

are necessary [111]. Elderly people may have problems with sight (i.e., they cannot read standard font sizes on mobile devices), hearing impairments, a reduced touch accuracy (e.g., due to tremor), or problems with memory. Although one of Shneiderman's eight Golden Rules says that working memory load should be minimized [308], this is not respected in many user interfaces.

Current "senior phones" try to target elderly people by drastically reducing features (following the "keep it simple" approach), arguing that seniors are overwhelmed by complexity or do not need all of them anyway. While this might be true for parts of the current generation, the future generation of seniors will be familiar with smartphones. They want to use the same or similar applications as before, or have even the need for specialized, new applications that address their special conditions, illnesses, etc. Consequently, the possibility to install new applications, update the software, and adapt the user interface is important. Hence, while maintaining or even extending functionality, the *interaction* with the devices and applications must be leveraged to address their limited capabilities and the problems listed above. We see here a great potential towards usable systems for the special needs of the target group of seniors. Multimodality, with the advantages we have presented in Section 2.2.1, can help improve the interaction in this direction. Lorenz and Oppermann suggest that interfaces for elderly people should contain redundancy [194], which could be well realized through multiple modalities.

Fitness and Sports

Fitness and sports applications are settled in the area of health-related applications as a subdomain, targeting users of all ages. The integration of health data in a central way to mobile operating systems (Samsung S Health[29], iOS HealthKit[30]) is an indicator for the increasing importance of tracking training data and vital signs.

Sensor-Based Activity Recognition

One important prerequisite to support training is the detection and identification of different types of physical activities. Activity recognition has been successfully employed in a variety of use cases [16, 94, 185, 200], using body-worn and pervasive sensors. The challenge of energy efficiency with mobile sensing has been tackled with approaches like Compressive Sensing [68, 69]. In medicine, quantifying qualitative aspects of human motion, such as motor performance, has been intensively researched, e.g., in the assessment of degenerative conditions such as Parkinson's disease [151, 236].

As a relatively novel application of pervasive computing, activity recognition has been applied in the sports domain. Augmentation of physical training devices with sensors has been used as basis for monitoring outdoor sports such as skiing [212] or tennis [3], as well as indoors, such as recognition and tracking of free-weight exercises with accelerometers in a glove [47]. In the gym, sensor data from balance board training [167, 298] has been used to provide feedback on the performance quality. Besides sensor-equipped training devices, body-worn

[29] `http://content.samsung.com/de/contents/aboutn/sHealthIntro.do`, accessed May 30, 2014

[30] `https://developer.apple.com/healthkit/`, accessed June 3, 2014

sensors have been used for the assessment of athletes in different disciplines, such as snowboarding [115], swimming [70], and running [317].

One early approach to integrate activity recognition with mobile devices was UbiFit Garden [57]. The application visualizes physical activity based on external sensors. For a certain amount of physical activity, flourishing flowers appear on the phone's display as motivational component. BALANCE [75] estimates the calorie expenditure in everyday life, contributing to long-term wellness management. Both smartphone solutions rely on sensors worn on the body.

Recently, a multitude of commercial health devices and sensors, such as oximeters and heart rate monitors, formerly reserved to professional use, are now available and can be connected to smartphones. GPS watches, pedometers, and heart rate monitors allow recording and tracking of physical activity. For home use, hardware platforms like Nintendo Wii or Microsoft Kinect encourage users to physical activity, yet without focus on correct execution. Activity loggers like activPal[31], FitBit[32] or smartwatches monitor health-related data and help create an activity profile. Samsung's Gear Fit[33] includes a heart rate monitor and offers personalized coaching instructions with the goal to increase motivation. Yet, those solutions build upon closed systems (e.g., Samsung's Galaxy Gear smartwatch only works with selected Samsung smartphones, and the Apple Watch only works with the iPhone), or rely on external sensor hardware.

Health and Fitness Apps

For an overview of the state of the art in commercial and free health and fitness support applications for end users, we conducted a heuristic evaluation [244] of 15 selected applications which were top-ranked in the Google Play Store[34] in the category *Health and Fitness*. Other marketplaces (e.g., Apple App Store[35], Nokia OVI Store[36]) contained the same or very similar apps, so that we argue that our comparative review represents the design space. We used the following heuristics, covering the most important aspects of health and fitness support:

- Utility and usability for regular training
- Instructional quality
- Usage of sensor data
- Motivational effect

In the following, we summarize the results of the evaluation. The detailed results can be found in our previous publication [169].

We found that all applications covered only a varying subset of important aspects. Based on the heuristic evaluation, we classified them into three categories: GPS trackers, workout planners, and exercise books.

[31] http://www.paltech.plus.com/products.htm, accessed May 9, 2014

[32] http://www.fitbit.com/, accessed May 9, 2014

[33] http://www.samsung.com/de/consumer/mobile-device/mobilephones/wearables/SM-R3500ZKADBT-features, accessed May 9, 2014

[34] at time of the evaluation called Android Market

[35] https://itunes.apple.com/us/genre/ios/id36, accessed May 31, 2014

[36] http://store.ovi.com, accessed May 31, 2014

GPS trackers are applications that record location traces for outdoor activities like running or cycling. If external sensors like heart rate monitors are connected to the phone, training instructions can be adapted to the heart rate. Further information from built-in sensors, such as accelerometer or magnetometer, are usually not used. With the advent of dedicated motion coprocessors as Apple's M7[37] in the iPhone 5s, this could likely change, as they allow to continuously log and process activity data (e.g., to count steps) with almost no additional battery consumption. The second category, workout planners, accompanies goal-directed workout, such as weight lifting or bodybuilding. Exercises are typically organized by body parts, so that the user can find suitable exercises to train particular muscle groups. While some applications support the user by counting repetitions, they provide no *qualitative* monitoring of the exercise performance. The third category usually offers the least interactive set of functionality, but the deepest background on the correct execution of exercises, why we call them exercise books. These apps often focus on yoga or gym exercises.

Based on our review of related work and app store content, we can summarize the state of the art in research and industry as follows:

- Both in research and industry, mobile systems have entered the sports and fitness domain. Commercial systems mostly confine to monitoring and tracking. Research is interested in activity recognition and classification, which has been demonstrated for different disciplines of sports. These approaches rely on external sensors, which makes the real-world setup complicated, depends on external hardware, and makes training not really "ubiquitous".
- Commercial apps focus on individual aspects (which is reflected in the categorization in GPS trackers, workout planners, and exercise books). Some logger apps support multiple activities, addressing the desire of comprehensively supporting fitness in different situations and locations. A further indicator towards a comprehensive approach is the trend to health apps like Samsung's S Health[29] or Apple's HealthKit[30] as part of iOS.
- Both research [57] and industry (e.g., uploading scores to social networks) use gamification [79, 82, 83] and competitive elements to increase motivation. However, no *intrinsic* long-term motivation, e.g., by individualized and personalized training feedback that actually can result in improvement, is created.
- All approaches do not focus on how the user interaction with a training application should be adapted for training situations (e.g., to be distraction-free, reduce the need for configuration, etc.).

2.2.3 University and Education

Universities as research institutions and think tanks often serve as testbeds for ideas and prototypic systems. Innovative approaches targeting the education and teaching domain itself are however slowly introduced. We give an overview on current developments in electronic, mobile and multimodal learning.

[37] `http://www.apple.com/iphone-5s/features/`, accessed May 31, 2014

E-Learning and Mobile Learning

The first association of using digital technologies in an educational context is *e-learning* (electronic learning), which is meanwhile an established complement to traditional offline learning in schools and universities. It is anticipated that e-learning produces better learning outcomes, as it allows *personalized learning*. The personalization makes learners to true subjects (not objects) and allows a greater diversity, participation and responsibility [142, 288].

In our work, we focus mainly on e-learning in higher education, i.e., in a university context. In 2006, Kim *et al.* [155] noted many opportunities for new learning methods in higher education facilitated through mobile technologies. They anticipated, e.g., a transition from traditional university courses to mobile learning settings, and online course management. Meanwhile, e-learning platforms like Moodle[38] are established to accompany courses. They provide online learning units, interactive content, and quizzes to verify what has been learnt. Online learning units can be integrated in courses (blended or hybrid learning [204]), but also serve for *distance learning*, e.g., for reworking content at home. A comprehensive overview of distance learning tools is provided by Garrison [110]. Massive Open Online Courses (MOOCs) [203] can even replace classic presence learning by providing all relevant material online through platforms like Coursera[39]. Even examinations can be taken via this platform.

However, learning platforms are not initially designed for *mobile* use. MLE (Mobile Learning Engine) was an attempt to use Moodle on mobile devices, either web-based or with a native application [135]. So-called MILOs (Mobile Interactive Learning Objects) contained chunks of information that were small enough to be consumed on mobile devices. They are also better suited for mobile learning which is likely to take place in short periods (waiting times, on the go, etc.). Users can realize mobile learning scenarios, e.g., location-based learning or uploading own material with the phone. However, MLE is not being further developed since 2009, so that a widely used or standardized mobile learning platform currently still does not exist. Holzinger *et al.* [134] suggested an extension to mobile learning objects called XLOs (X-Media Learning Objects). XLOs make learning content accessible on a greater variety of devices, e.g., MP3 players, PDAs or TVs, with the goal of "pervasive learning" (at every location, every time, cross-device).

The transition from (computer-bound) e-learning to mobile learning (m-learning) [307, 320] is still in its beginnings. As defined by Ally [4], mobile learning is "the delivery of electronic learning materials on mobile computing devices to allow access from anywhere and at any time". For an overview of mobile learning technologies, e.g., refer to Naismith *et al.* [241]. Mobile learning has reached momentum with the rise of small and light smartphone and tablet devices. Manguerra and Petocz [201] investigated how classroom settings could be enhanced through these technologies to match the expectations of students who grew up in the "digital age". Mobile devices allow, e.g., to work with lecture slides and course material (e.g., in PDF format) on the go, watch podcasts, and thereby facilitate time- and location-independent learning. Furthermore, mobile devices are a catalyst for learning forms such as e-books, polling, blogging or mind mapping [53]. They can be used to consume educational resources or even complete online courses, e.g., through iTunes University[40]. Beyond that,

[38] `http://www.moodle.org`, accessed February 28, 2014
[39] `https://www.coursera.org`, accessed May 31, 2014
[40] `http://www.apple.com/education/itunes-u/`, accessed May 24, 2014

digital market places enhance the functionality of smartphones and tablets, as they offer a plenitude of apps for specific learning tasks and contexts. As of July 2014, more than 120,000 apps in the *Education* category could be found in Apple's App Store[41].

Mobile learning also allows novel and more pervasive ways of learning. The importance of context awareness and contextual adaptation has been recognized by several researchers [202, 310, 323, 337]. For example, Soualah-Alila *et al.* [310] investigated how the most appropriate learning content can be presented to learners, based on a semantic level (characterization of learning content and learner), and on a behavioral level (potential information overload due to the learner's context). Looi *et al.* [193] explored how mobile learning can provide access to authentic contexts (learning in the wild) and couple physical actions with cognitive activities. This kind of experience-based learning can be applied where practical skills are required, e.g., in the medical area [134, 307], in apprenticeships [320], or in lifelong learning [262]. Thüs *et al.* [323] give an overview on further frameworks for mobile learning in context. Ogata *et al.* presented a ubiquitous learning log [251] as digital record of what has been learnt in the daily life with ubiquitous technologies. The rising interest in mobile learning is reflected in emerging journals dedicated to this topic, e.g., the *International Journal for Mobile Learning and Organisation (IJMLO)*[42].

For instructors and docents, the rapid emergence of e-learning technologies is a challenge. They need to fathom the range of possibilities and decide what is reasonable and adequate for their teaching. The risk is that unreflected adoption of digital possibilities replaces didactically well-grounded preparation of courses [86]. Ramsden [269] argues that teachers still need profound didactic knowledge to develop learning concepts and to structure their courses, be it offline, online or any combination of them. Especially in higher education, courses or tutorials are not held by full-time lecturers, but by associates without explicit didactic education, or by Ph.D. students [346]. One possibility could be to use mobile learning also for teacher training, as presented in a project by Seppälä and Alamäki [304]. We will present an own approach in this direction in Chapter 4.

Multimodal Learning

Moreno and Mayer define multimodal learning environments as "learning environments that use [...] different modes to represent the content knowledge" [237], in their case verbal and visual representations. It is argued that multimodal learning, as it involves different senses, produces more substantial learning effects, be it because the memorization process is more effective, or because learning is more playful and thus more fun [133, 182, 333].

In our context, we are especially interested in (novel) learning modalities enabled through technology. In prior research, physical objects have been enhanced with digital technology to so-called "smart" [172] or "tangible" objects. They allow situational and playful learning by experimentation, combined with the advantages of e-learning. Schmidt *et al.* [298] developed with the SensorVirrig a cushion with integrated ball switches, a compass and a pressure sensor, usable to control objects in learning games. The Display Cube by Kranz *et al.* [322] is augmented with accelerometers and small screens, allowing to answer multiple-choice tests

[41] http://www.pocketgamer.biz/metrics/app-store/, accessed July 31, 2014
[42] http://www.inderscience.com/ijmlo, accessed May 12, 2014

by physically rotating the cube. Both objects are examples for multimodal and playful interaction devices for kids. Besides "pervasive learning" at home or in other places, such smart objects could also be used at schools or universities to enhance and complement traditional lessons.

Learning and Teaching Environments

Opening up the design space, mobile application support in the educative context not only covers e-learning applications, but also services and tools in learning and teaching environments. This includes tools for docents and docent training as well.

Many universities, as representations of educative environments, offer various digital services to students, such as course management systems, campus maps, or internal news feeds. These are more and more available in native apps for mobile devices, such as iLancaster[43], MIT Mobile[44] and the CMU App[45] (just to name a few examples). Voting systems [309] are used in teaching to verify comprehension, foster discussions, or create a feeling of community.

Abowd [1] presented with Classroom 2000 at Georgia Institute of Technology one of the first projects to holistically support learning with mobile devices. It includes both instrumented rooms supporting lecture capture and mobile personal interfaces (tablet PCs), which is used for live-annotating lecture slides. In a further step, rooms were instrumented with cameras, microphones and electronic whiteboards as "living laboratory". With electronic whiteboards, many of the back then prototyped possibilities are now off-the-shelf available, but integration with the student's personal mobile devices has still not been achieved.

A true integration of mobile systems and infrastructure systems on campus in the sense of an intelligent environment [279, 282] has rarely been pursued in the educative context. This research direction is summarized under the vision of ubiquitous learning (u-learning), which is described by van't Hooft *et al.* [330] as learning in an environment where "students have access to a variety of digital devices and services, including computers connected to the Internet and mobile computing devices, whenever and wherever they need them".

Beyond the classroom, pervasive displays, public terminals or door signs [132, 280, 299] could be used as further interaction points. Berg *et al.* [23] suggest a stronger integration of social networks and campus services. Wheeler and Waggener [344] outline the potential of cloud computing for new services on campus. At the end of Chapter 4, we will portray a teaching and university scenario featuring an interplay of different applications and interaction modes, subsumed under the notion of MUSED interaction.

2.2.4 Indoor Navigation

In Chapter 5, we will look at indoor navigation as an example domain for multimodal interaction. Indoor navigation is a special case of pedestrian navigation [49], posing particular

[43] `http://ilancasterinfo.lancs.ac.uk`, accessed February 27, 2014

[44] `https://play.google.com/store/apps/details?id=edu.mit.mitmobile2`, accessed February 27, 2014

[45] `http://www.cmu.edu/cmuapp`, accessed February 27, 2014

requirements to localization technologies. Therefore, we first give an overview on approaches to estimate the position of a mobile device, using the built-in sensors (a list of available sensors was provided in Section 2.1.2). We then present vision-based localization, the method we chose as a basis for our indoor navigation approach, and outline its advantages and challenges. Subsequently, we give an overview on current user interfaces for pedestrian navigation systems.

A commonly used outdoor localization method for mobile devices is satellite localization using GNSS. Alternatively, cellular or WLAN localization [191] can be used if GNSS is not available (e.g., when no satellites are visible due to the "urban canyon problem", or indoors). By signal multilateration between the device and GSM cell towers, the position can be coarsely estimated (in the range of a few dozen to several hundred meters). Liu *et al.* [191] discuss wireless localization techniques and classify them in triangulation-based, proximity-based and fingerprinting approaches. Fingerprinting is a technique that positions a device based on the proximity to known WLAN networks [49]. Therefore, a reference database of access points is necessary as, e.g., offered by Google[46] or Skyhook[47].

There are a number of approaches specially dedicated for indoor localization. They are motivated mainly by two facts. First, traditional outdoor methods like GPS work indoors only to a limited extent as the signals are too weak and potentially shielded, e.g., by metallized heat insulation windows (though there are approaches to improve GPS reception indoors [73, 294]). Second, the requirements on localization accuracy are higher indoors [207] (in the range of a few decimeters instead of meters outdoors), why most indoor approaches go a different way. They are based on different technologies and sensors available in mobile devices, such as Infrared [40], DECT [164], inertial sensors [5], Bluetooth (e.g., with the upcoming iBeacon[48] technology), or a combination of multiple sensors, e.g., WLAN for approximate localization and IMU for accurate relative positioning [127, 348]. Also the camera has been used in different ways. The location can be determined by explicit markers [239] or by image matching using feature extraction [105]. In addition to estimating absolute locations, it is also possible to compute relative locations using visual odometry [128], e.g., by tracking features over time [158].

For a more extensive discussion of indoor positioning and path planning techniques, we refer the reader to Fallah *et al.* [96].

Vision-Based Localization

The user interface we will present in Chapter 5 is adapted to vision-based localization, i.e., using the vision modality as primary source for location estimation. We here motivate why we decided to build upon this localization technique.

Visual localization works very similar to the orientation of humans: Visual perception is used to compare the environment to known information. Similar to humans, who need to have

[46] http://static.googleusercontent.com/media/www.google.com/de//googleblogs/pdfs/google_submission_dpas_wifi_collection.pdf, accessed February 24, 2014

[47] http://www.skyhookwireless.com, accessed February 24, 2014

[48] http://support.apple.com/kb/HT6048?viewlocale=en_US&locale=en_US, accessed May 31, 2014

visited a place earlier in order to know it, a vision-based localization algorithm needs a reference database of images of the environment with associated location information with each image. The localization procedure is as follows. The user records query images with the camera of the smartphone, which are sent to a server. A similarity search with the reference database is conducted, and the location and orientation of the most similar image is used as an estimate for the user's actual position. The comparison is performed based on characteristic properties of the image, so-called *features* (e.g., SIFT [195], MSER [206], SURF [21], FAST [286], BRIEF [41], or ORB [287]). If these feature are size- and rotation-invariant, the matching algorithm also works when query and reference image have not been recorded from exactly the same angle and distance. From the displacement between the two images, the exact orientation and location can be accurately estimated [302]. In the remainder of this paper, when the term *visual localization* is used, we explicitly mean the above described method. This approach, which is also referred to as content-based image retrieval (CBIR), stands in contrast to other localization methods which likewise use the camera, such as, e.g., marker-based approaches [239].

Besides the fact that visual localization is a good example for a sensor-driven and multimodal system, there are further reasons why we chose this method.

Infrastructure-less Localization

Visual localization works without a special infrastructure. Other active localization methods (i.e., where the device localizes itself actively) need an augmented environment, such as electronic tags (e.g., iBeacons[48]) or a dense coverage of WLAN access point infrastructure. Especially in large-scale environments, such an augmentation is expensive and effortful to establish. Feature matching further has the advantage that exact localization can be performed at any location, while infrastructure-based (e.g., marker-based) approaches only work at certain augmented areas. Apart from these "key locations", the position needs to be estimated with relative positioning techniques, such as dead reckoning.

No Special Hardware Requirements

A common camera-equipped smartphone is sufficient for visual localization. Thanks to fast multi-core processors and high-quality cameras, state-of-the-art devices are powerful enough to perform image recognition and analysis in real-time. By contrast, approaches based on signal metrics, such as angle (AoA), time (ToA), or time difference of arrival (TDoA) require special hardware, such as directional antennas or ultra-accurate timers [189].

High Accuracy

With a database of sufficiently densely recorded reference images, visual localization can reach up to centimeter-level accuracy [302]. Based on the position of feature points, even the pose (i.e., the viewing angle) can be detected. This can be achieved, to some extent, with other approaches as well, e.g., by exploiting the Doppler effect [11] or MIMO (Multiple Input Multiple Output) [351], but not as accurate as with image matching. However, the image database must be built up in a one-time effort (by mapping the environment) and updated regularly when buildings and objects therein significantly change. Approaches based on signal strength measurements reach only a one-meter accuracy even in laboratory tests

[187]. In the real world, where the typical density of access points is mostly lower, expected localization accuracies are likely to be inferior to those in controlled experiments.

However, localization using vision entails some challenges. First, it requires reference data, i.e., the environment must be known a priori in order to later localize the device within the environment. Reference images must be gathered in the first place and the exact location must be assigned to each image, e.g., by using a mapping trolley as presented in [138]. Since the environment could be subject to change (e.g., when shop window displays, adverts or posters are replaced), a way to update the reference material must be foreseen. This can be done centralized or in a collaborative approach, where query material is tagged with a location manually by users and eventually becomes part of the new reference dataset.

There exist several implementations of camera-based location recognition systems [125, 239, 302, 343]. Hile and Borriello correlated a floor plan or a previously captured reference image to estimate the device's pose and to calculate an information overlay [125]. However, the system only works for static images. Mulloni *et al.* [239] relocalized a phone by recognizing visual markers and displayed the new location on a map. Werner *et al.* [343] and Schroth *et al.* [302] presented localization approaches through feature-based image matching, but without specific focus on user interfaces.

User Interfaces

In this dissertation, we focus on user interfaces for indoor navigation, and the employed modalities for interacting with the system. Route guidance can be provided using different channels, which all have their individual advantages and drawbacks, as outlined by Kray *et al.* [173]. In the following, we discuss several modalities and user interface elements employed in existing pedestrian navigation systems.

Textual Instructions

Text instructions can be given in written form, spoken (e.g., using a text-to-speech engine), or both. Depending on how familiar the user is with the environment, the amount of information can be reduced or increased. To reduce the user's cognitive load, it can be beneficial to combine directional information with street names or landmarks (e.g., "turn right at the elevator"). In most cases, text instructions will be verbalized, which has the advantage that no visual attention is required. However, for spoken instructions to work, the system must have a very accurate location estimate, compared to some forms of graphic visualizations [173].

Two-dimensional Graphics

In the most abstract form, this graphic visualization consists only of an arrow indicating the walking direction. This high degree of abstraction can be an advantage as it reduces distraction. On the other hand, it can be disadvantageous, as additional cues for orientation are missing. Such additional cues can, e.g., be given by outlining alternative paths, or by labels next to the arrow [173]. Such simple visualizations are well suited for small displays and could also be shown on the limited screen size of newly emerged accessory devices like smartwatches or head-mounted displays (HMDs).

Two-dimensional Maps

A visualization with more details than abstract 2D graphics is a geographic 2D map, annotated with route information [173]. Maps integrate context in a very natural way, in contrast to the previously presented instruction types. The user not only gets information on the route, but also on what is close to the route. Such contextual cues foster exploration (finding objects of interest nearby), and can be helpful for self-orientation. This is known from tourist maps where landmarks [214, 285] are drawn as small pictures directly in the map. To further save cognitive resources, maps can be aligned on the screen with the user's looking direction, so that no mental rotation of the map has to be performed [173]. On the other hand, it is important to use a reasonable level of detail: While a low level of detail might not contain sufficient information, too much information might confuse the user. Meilinger *et al.* [210] found that users could self-localize faster with schematic maps than with true-to-scale maps.

Butz *et al.* [40] investigated how to adapt route instructions according to the quality of location and orientation estimate. They basically recommend to increase the level of details the worse the system can localize the user. Then, the user can use the additional information to self-localize. Butz *et al.* argue that the visualization can simply consist of a directional arrow if the accuracy is well, but as soon as the orientation of the user cannot be reliably estimated any more, it must include landmarks (e.g., elevators or staircases), and indicate the north orientation. Instructions should then be formulated in a way that they cannot be misconceived (e.g., instead of "turn left", they could say "turn until the staircase is to your right"). It is also suggested that the user can specify the own position manually to dissolve ambiguities. Another way to account for unreliable location estimates in a map view is by visualizing the location not with a point, but a circle with varying radius [20].

Three-dimensional Elements and Landmarks

Using the third dimension can make the view more realistic. A pseudo-realistic 3D view can simulate the real environment with model character. Significant objects with "landmark" character can be integrated to ease orientation for the user. Kray *et al.* [173] suggest that not all parts of the real environment have to be represented in the same level of detail and distinguish four levels of detail: visible, distinguishable, classifiable, and identifiable. Objects that are important for orientation are depicted more recognizable (e.g., rendered with more triangles or using a realistic texture), while for less important objects a lower level of detail is sufficient. The stronger a visualization resembles the real environment, the easier the user can match the provided output to the real world. Using "visual search", users can match their position based on the rendering, which can accommodate inaccuracy. The "fun factor" should likewise not be underestimated [249]. In a comparative study [173], a 3D visualization was more popular with users than a map interface, although initial orientation was easier with a 2D map. The authors also found that users like to interact, instead of passively "consume" navigation instructions.

Acknowledging that landmarks are helpful for orientation, some researchers focused on images (as detailed as possible) as primary means for navigation. Users can then determine their location by comparing the image material with the real world. To extend the amount of information, not only individual images, but panoramas can be shown. One of the most well-known systems of this kind is probably Google Street View[49]. Miyazaki *et al.* [218] generated panoramic views of the surrounding to give additional information, e.g., on buildings. Mulloni *et al.* [238] investigated in which perspective panoramas are optimally shown. They found that by top-down and bird's eye views of a panorama, users were quicker to locate objects in the environment than using a frontal view.

Haptic Feedback

The haptic channel can be beneficial in situations where visual or auditive feedback is not appropriate. Bosman *et al.* [31] presented a wrist-worn system targeted at visually impaired users. Wrist bands worn on the left and right arm vibrate to indicate the direction in which the user has to go, and could also be imagined for noisy environments. *HaptiMoto* [264] is a wearable motorcyclist navigation system based on haptic feedback. The authors use a tactile vest with vibration motors in the shoulder area, since the motorcycle's vibration makes navigation signals unnoticeable in wrists, hips and waist region. According to the "shoulder tapping analogy", Prasad *et al.* [265] found that users interpret short tactile signals on the shoulder as "tapping", i.e., as a request to turn into this direction, while a longer-lasting vibration (more than one second) "pushes" subjects to the direction of the other shoulder.

Augmented Reality

Augmented Reality (AR) superimposes virtual elements over a live camera view. This way, users do not need to translate between the virtual representation and the real world [242]. AR has been used in manifold ways. For an overview, refer to, e.g., two surveys by Azuma *et al.* [12, 13]. For pedestrian navigation, AR has been used in different ways to provide navigation instructions. A straightforward way is projecting directional arrows on the floor [190, 334], but also more playful ways have been investigated. Penguin NAVI[50] uses AR-projected penguins that walk in front of the users to guide them to Tokyo Aquarium. In a comparative evaluation by Walther *et al.* [334], users attested the AR-based system a better usability than a user interface relying on maps. Miyashita *et al.* [217] employed AR visualizations for a museum use case. The exhibits were implicitly used as visual markers to define the path through the exhibition. When visitors spotted the exhibit following next on the intended path, it was augmented with additional background information.

Multimodal Instruction Presentation

Ariwaka *et al.* [9] argue that multiple channels to provide information do not only reduce errors, but also increase users' confidence and trust in the system. Consequently, researchers experimented with combining modalities or offering users a choice of alternative ways to get guidance instructions. Hile *et al.* [126] combined textual route description with images of distinctive objects, which serve as additional cues for self-orientation. A similar approach is presented by Beeharee and Steed [22]. Liu *et al.* [190] used a set of different modalities (images, audio, and text) to match the requirements of cognitively impaired users. They found that participants' preferences for modalities widely varied, but that they generally appreciated the combination of modalities.

[49] https://www.google.com/maps/views, accessed May 31, 2014

[50] http://gizmodo.com/every-gps-app-should-make-you-follow-an-adorable-pack-o-1441939981, accessed June 12, 2014

2.3 Designing and Implementing Multimodal Systems

We now give an overview on approaches to model multimodal interaction, and on existing software toolkits as well as frameworks that help create multimodal and context-aware applications. We see context awareness as a prerequisite for many multimodal systems, as we will argue in this section.

2.3.1 Modeling Multimodal Interaction

Prior to the actual implementation of a multimodal system, there are tools and formalisms to model the behavior of the system. The CARE properties, initially proposed by Coutaz *et al.* [63], are the basis for many multimodal toolkits and design approaches that we will describe later. CARE is an acronym for Complementarity, Assignment, Redundancy, and Equivalence, which describe the roles and relations of modalities involved in a system.

Furthermore, multimodal interaction can be modeled on an abstract level with modeling languages. Several notations have been proposed to this extent, for example XISL (eXtensible Interaction Scenario Language) [150], UsiXML (User interface eXtended Markup Language) [311], NiMMiT (Notation for MultiModal interaction Techniques) [329], MIML (Multimodal Interaction Markup Language) [8], SMUIML (Synchronized Multimodal User Interaction Modeling Language) [88], and M4L (Mobile MultiModality Modeling Language) [92]. We will not discuss the particularities of each of these markup languages in detail; for a brief comparison of the most important ones, we refer to Dumas *et al.* [88]. For our framework we will present in Section 6.3, we do not rely on a formal modeling language, but focus on rapid prototyping of multimodal behavior, similar to Dey *et al.* [76] in the domain of context-aware computing.

2.3.2 Software Toolkits and Frameworks

To support the actual implementation of multimodal systems, several toolkits and frameworks have been presented. Let us at this place point out the difference between the term “toolkit” and “framework”: While a toolkit is a loose collection of tools (constructs, routines, algorithms in software terms) that can be called by own code, a framework is a reusable “body” of software that calls self-implemented code to produce custom applications. This is often referred to as “Hollywood principle” (“don’t call us, we call you”) [97, 306].

Toolkits and Frameworks for Context Awareness

The choice of beneficial modalities often depends on the user’s context, such as time, location, or the social setting. Manifold toolkits for creating context-aware applications have been presented in research. As probably one of the pioneer works, we here mention Dey *et al.*’s Context Toolkit [76], even though it was implemented in Java and not explicitly focused on mobile device usage. Other approaches are Context Studio [160], Context Phone [268], SeeMon [148] and MobiCon [181]. These toolkits abstract context sources, provides modules that developers can include in their own code, while focusing on different aspects. Kang *et al.* [148] emphasize the energy efficiency of their context monitoring by redundancy avoidance. Lee *et al.* [181] put the focus on sensor-rich mobile environments and anticipate the detection of contexts like activity recognition, sports, or weather by using external sensors. Hence, their framework is not intended for autonomous use on mobile devices. Schuster *et al.* present a context-oriented programming (COP) extension on code level [303]. With special language constructs, context-dependent code variations are defined using layer definitions and partial methods. This approach requires a modified compiler and is thus less flexible in its use compared to traditional framework-/toolkit-based approaches.

All these toolkits have in common that they focus on context awareness and do not explicitly support the implementation of multimodal interaction. We, as one contribution of this work, explicitly support the implementation of multimodal behavior with our framework presented in Section 6.3. The usefulness of many context toolkits might have become limited with the advent of today's mobile operating system's software development kits (SDKs) that already provide numerous possibilities to retrieve (simple) context information. However, the SDKs' functionality usually does not go beyond a simple sensor–context mapping and does not provide advanced context inference (e.g., based on machine learning). The effort to actually implement interaction modalities based on this context information would still be significant, as context toolkits do not offer special support for that.

Toolkits and Frameworks for Programming Multimodal Behavior

To this end, some frameworks and toolkits emerged that actually have the goal of creating multimodal interfaces. We will present some of them and point out their limitations in order to differentiate them from our own mobile multimodal interaction framework, which we will describe in detail in Section 6.3.

Krahnstoever *et al.* [162] built a framework using speech and gestures for natural interaction with large screens. Their approach has limited generalizability, as it is focused and adapted to the large screen use case and the chosen modalities. Bourguet [32] presented a toolkit for testing multimodal interface designs (in their work, it was used to prototype multimodal drawing applications). The interaction scenarios can be modeled using finite state machines. The system consists of a graphical user interface builder and the execution framework itself. While the approach is interesting for experimenting with modality combinations, it is developed for the PC and not targeted at mobile platforms. Flippo *et al.*'s framework [101] has likewise the goal of rapid prototyping of multimodal interaction. A special focus is the fusion of different modalities, which is realized by a frame-based approach and a semantic parse tree. Also Flippo *et al.*'s work was targeted at and evaluated with a desktop use case, using a mouse and speech input combination. With *SwingStates*, Appert and Beaudouin-Lafon [7] present a system to facilitate the development of novel input methods (e.g., multi-handed or pressure-sensitive input) by a state machine. It is an enhancement to the Java Swing toolkit and, therefore, likewise targeted at desktop applications. Cutugno *et al.* [66] present an architecture to fuse different events so that they are interpreted as one single intention. Multiple so-called input recognizers are bundled by an "interaction manager"; the behavior is evaluated by a non-deterministic finite automaton. Dumas *et al.* [89] provide a comparative review of further multimodal toolkits, and differentiate them by architectural characteristics, reusability, and further aspects.

While the previously presented systems address multimodal input, there are, on the other hand, so-called tasking applications [273] which automate workflows based on different events. For example, they utilize context to change output modalities (e.g., to mute ringtones in a silent environment). Applications like Llama[51], Locale[52], or Tasker[53] that have appeared in application stores can be counted to this category. The emergence of such commercial automation applications confirms an existing interest in engagement with multimodality.

Code in the Air by Ravindranath *et al.* [273] supports the development of tasking applications by a task execution framework that distributes and coordinates tasks. It features a server-based task compiler which allows the implementation of "cloud tasks" (e.g., track the location of a mobile device and store it on the server), and the integration of multiple devices. However, mobile devices only have a task runtime which makes the system not fully autonomous.

Elouali *et al.* [92] propose MIMIC (MobIle MultImodality Creator), a graphical system to model multimodal interfaces integrated in the Eclipse[54] environment. Based on modeled application screens and input and output interaction events, MIMIC can generate Android code. While Elouali *et al.*'s system shares some ideas with the M3I framework we present in Section 6.3, it offers less flexibility to developers, as only a predefined library of interaction events can be used for building own applications.

While this comprehensive overview does not claim to be complete, it shows the state of the art of toolkits for context acquisition and for implementing multimodal interaction, using diverse approaches, including state machines [7], state charts [329], automatons [66], petri nets [256], or rule-based evaluation [273]. In summary, there is currently to our best knowledge no system that fulfills *all* of the following criteria that are met by our solution. We provide a holistic framework that

- improves modeling, prototyping and implementing of multimodal behavior
- is flexible and applicable in any development environment (no special compiler, software, etc.)
- is targeted at state-of-the-art mobile operating systems
- runs autonomously, without server-based evaluation or control through the cloud
- supports both input and output
- focuses on multimodality, not only context
- integrates an approach that supports the human mental modal of multimodal behavior

However, few insights exist on actual modality usage, preferences with regard to modality switches, enhanced possibilities for input and output, and awareness on modality settings. We investigate research questions towards this direction in Chapter 6.

[51] `https://play.google.com/store/apps/details?id=com.kebab.Llama`, accessed March 12, 2014

[52] `https://play.google.com/store/apps/details?id=com.twofortyfouram.locale`, accessed March 12, 2014

[53] `https://play.google.com/store/apps/details?id=net.dinglisch.android.taskerm`, accessed March 12, 2014

[54] `https://www.eclipse.org`, accessed May 13, 2014

2.4 Evaluating Multimodal Systems

There exists a multitude of evaluation methods, which can of course also be applied to multimodal systems. In this section, we will give an initial overview of the most important evaluation methods, where we will at some places already indicate why this method is advantageous for the use with multimodal systems. In Chapter 7, we will then investigate the characteristics of selected methods, and report based on our own experiences.

2.4.1 Evaluation in the Laboratory

Expert Review

An experienced researcher or domain expert (e.g., a clinician in case of a medical application [169]) tests and reviews the system. The review can either be according to the professional experience, or based on strictly defined criteria (so-called heuristics). The latter method is called *heuristic evaluation*, as one form of expert evaluation [244]. One of the most frequently used sets of heuristics are the ten usability heuristics defined by Nielsen [244].

Focus Group

A focus group [259] is a structured interview with a small number of participants. Its goal is an initial exploration of goals and needs of potential users of a system. Through the group setting, a large amount of interactivity is given, which can "stimulate participants to raise issues that they might not have identified in one-to-one interviews" [177]. Often, focus group discussions are video-recorded for the later evaluation. Focus groups provide qualitative data unlike, e.g., laboratory experiments (see next paragraph).

Laboratory Experiment

In a laboratory experiment, test subjects typically perform one or multiple tasks involving a prototype of (parts of) the system which is evaluated. Often, different manifestations of the prototype are compared (e.g., visualizations, user interfaces, implementations) with regard to different parameters (e.g., [30, 116, 290]). It can, for example, be measured whether the manifestations have an effect on efficiency (task completion time), effectiveness (task completion, error rate), task load, or satisfaction. Measurements can be repeated in different conditions using the same subjects (within-subjects design), or different groups of subjects participate in different conditions (between-subjects design) [36]. Subject of evaluation may be conceptual art [113], mockups [226], or functional prototypes [30, 221]. For multimodal interaction, often a certain level of maturity is required for a reasonable evaluation. If this maturity is not yet given, the functionality can be simulated using the Wizard-of-Oz approach (WOz) [152]. In this technique, a researcher is responsible for the output of a system, rather than the system itself – in other words, he plays the "wizard" who makes the system work. This allows testing interaction methods before implementing them. In Section 7.2, we discuss the usage of WOz in the context of multimodal interaction.

2.4.2 Evaluation in the Field

As MUSED systems often interact in a particular way with their environment or are context-dependent, some aspects (e.g., which modalities users choose in which situation) can only be evaluated in the field. We distinguish three common types of field studies by the goal they pursue [36]: studies of current behavior, proof-of-concept studies, and experience studies using a prototype.

Studies of Current Behavior

Studies of current behavior investigate the status quo: by observing habits, practices and workflows, studies of current behavior try to gain a better understanding of users and how they use technology in their lives. This proceeding is also referred to as ethnography [177] and has an exploratory character, similar to focus groups in the lab. Since the findings shall inform the further design of a prototype, such studies will typically be situated at an early point in research. While in a focus group participants need to reflect on their behavior and be aware of their habits, observation allows an unbiased view on actual phenomena and thus leads to more diverse observations [321].

Proof-of-Concept Studies

Proof-of-concept studies bring prototypes to the real world, in order to investigate whether they function as intended or actually have the desired effect (e.g., shortening the time to accomplish a certain task). Such aspects can usually not be investigated in a laboratory experiment. However, often only very specific research questions are investigated, so that the proof-of-concept study is of short duration, where a researcher is present all the time. A special case of an investigation of a phenomenon under real-life conditions is the *case study* [349]. Here, it is tried to investigate a scenario in a holistic way by using quantitative and qualitative methods. Findings can add to theory building and often inform follow-up experiments [349]. Case studies have, e.g., been conducted for the evaluation of multimodal map interaction [54], musical interfaces [153], or privacy concerns with location-aware systems [18].

Prototype Experience Studies

The third type, experience studies, are often run in the long term, i.e., over several weeks or months [59]. In contrast to proof-of-concept studies where it is investigated *if* a system works, long-term studies have the goal to gain insights *how* a system is adopted and used in people's daily lives [263]. For long-term studies, it is not possible that researchers are permanently present to make observations during a longer period of time. Therefore, data collection methods are required to acquire that information. Selected methods we consider particularly important are discussed in the following.

Data Logging

With *data logging*, a device collects data or context information automatically and without user intervention [106]. Examples are all sorts of quantitative measures like usage data of applications, or fine-grained context information. Logging has, e.g., been employed to quantitatively investigate interaction with mobile devices [95], identify usage patterns [25], or for life logging [149]. Often, information gained by logging would otherwise be impossible to gather, be it regarding effort or time [243]. In addition, logging mostly happens unobtrusively in the background so that subjects do not even notice it. This is important for data validity, as subjects then cannot change their habits because of feeling observed [144], which is known as *Hawthorne effect* [208]. The problem of altered behavior due to experimental conditions is further discussed in Section 7.4.

Kärkkäinen *et al.* [149] point out that logging bears the risk of being perceived as privacy-threatening by subjects. Researchers should therefore carefully consider which data is actually necessary to be logged. With the logging tool SERENA which we present in Section 7.5, we account for best-possible privacy, as the researcher can confine logging to only the data that is relevant for the study. As a side effect, later evaluation is potentially simplified as the amount of data to be processed (and thereby the “needle in a haystack” problem) is reduced.

What cannot be captured using data logging is qualitative feedback, such as users’ intentions [106]. In order to interpret the logged data, additional techniques are often required. Two of them, experience sampling and the diary method, are explained in the following.

Experience Sampling

Stemming from psychology, the *experience sampling method (ESM)* is an approach where participants actively collect in-situ data [44, 257]. These can be samples from daily life or certain situations, e.g., photos, audio, or videos, but also qualitative feedback like annotations and comments that describe the subject’s thoughts, or responses to questionnaire sets. If questionnaires are used, sets should contain a small number of items in order not to disrupt users and to keep the additional effort low. The practical application of experience sampling has been described, e.g., by Christensen *et al.* [51]. ESM requests can be pre-scheduled (either randomly or time-based), or triggered by specific actions of the user [36, 56]. The latter way allows to get qualitative insights, e.g., about the circumstances when a subject used an application. Consolvo *et al.* [56] used surveys on mobile phones for context-triggered experience sampling. The approach of *Mobile Probes* [139] offered the additional possibility to include images to experience sampling records.

Diary

While the moments for collecting data are triggered in experience sampling, the *diary* method allows users to choose this moment on their own [44, 177, 197, 257]. Diaries can be unstructured or structured [177, 257]. Due to the high degree of freedom participants have, diaries bear the risk that users forget about reporting or that they do not remember events correctly. Forgetfulness is mentioned as problem by Lester *et al.* [185]. Other distorting factors of self-reports (not only applying to diaries) are recency effects [91] or intentional misreporting [185]. However, a positive effect of self-determined diary entries is that researchers implicitly learn about the importance of events to subjects. An overview of diary designs is given by

Bolger *et al.* [27]. In the context of HCI, diary studies have, e.g., been used for investigating task switching and interruptions [67], collecting phone call data using voice mail diaries [257], analyzing random encounters in the context of location-aware computing [55], or investigating the usage of videos on mobile devices [139].

Mobile Data Collection and Research in the Large

In the field of Mobile HCI, where the subject of investigation is related to mobile interaction, using the very same devices for self-reporting is a plausible idea. Hufford and Shields [137] revealed in a meta analysis that electronic diaries result in a higher compliance with subjects and in better results. Recently, several dedicated research applications with the primary goal of data collections have emerged. As examples shall be mentioned here MyExperience [106], Momento [45], droid Survey[55], SurveyToGo[56], and EpiCollect[57]. These survey tools allow experience sampling or diary-based data collection. Some additionally log diverse kinds of usage information in the background, such as MyExperience [106] or AWARE [98].

On the other hand, there is the recent trend of using mobile applications deployed in app stores as vehicles to collect research data, under the buzzword of “research in the large” [28, 64, 123, 124, 171, 215, 292, 293]. Free apps offering a value or benefit to users (e.g., games [123] or useful tools [293]) attract a large number of users who provide a representative dataset for evaluation. In Section 7.5 we will discuss research in the large in further detail.

2.4.3 Surveys

In surveys, subjects are explicitly asked for their opinion, on the basis of (online or paper and pencil) questionnaires. While short sets of questions within the experience sampling or diary method can be interpreted as a survey, the term is usually used for longer sets of questionnaires, which are often employed before (pre-survey) or after a study (post-survey) [36]. They can be used in conjunction with both laboratory and field studies. A survey usually consists of different question types, such as open-ended questions, single or multiple choice questions, or Likert questions where subjects indicate their agreement to predefined statements on a scale [36]. From the data collected with surveys, researchers can gather qualitative insights (e.g., by analyzing the responses to open-ended questions) or quantitative results (obtained by applying statistics to, e.g., Likert responses).

[55] `https://www.droidsurvey.com/`, accessed March 12, 2014

[56] `http://www.dooblo.net/stgi/surveytogo.aspx`, accessed March 12, 2014

[57] `http://www.epicollect.net/`, accessed March 12, 2014

Part II

Multimodal and Sensor-Driven Interfaces in Different Application Areas

Chapter 3

Health, Fitness, and Activities of Daily Living

3.1 Problem Statement and Research Questions

A major goal in the field of mobile health (mHealth) is to help people to adopt and sustain a healthy lifestyle [157] with technical support of mobile devices. This research area, including self-monitoring, medication reminders, and fitness applications, has recently gained attention (also under the term "Quantified Self"), not only because of an increasing health consciousness, but also because of the aging society. In this chapter, we inquire into two applications in the area of activities of daily living (ADL) and personal fitness.

Activities of Daily Living

One effect of the aging society are problems with daily tasks, including medication intake. Due to the phenomenon of multimorbidity [327], elderly people often have a hard time to keep an overview of the number of pills they need to ingest, and patient information leaflets (PILs) are not always at hand, or too small to read. We argue that mHealth applications are suited to address these problems, since the best agers of tomorrow become more and more technology-affine, and thus able and even demanding to use smartphones and apps. However, there is a requirement for adaptations and for a stronger focus on usability to address the needs of elderly people who are, e.g., impaired in vision or have limited motor skills. We intend to address this problem by novel multimodal interaction methods for object identification, which will be detailed in Section 3.2.

Personal Fitness

Today's sedentary and busy lifestyles can lead to chronic conditions like obesity, diabetes or heart diseases [157], so that a stronger promotion of physical activity is important. Research in behavioral science shows that people are stronger motivated by short-term incentives than by long-term goals [103]. However, positive effects of physical activity often show only in the long run (e.g., muscle growth, improved condition, weight loss). Therefore, one important measure is to maintain long-term motivation.

We currently observe an emerging trend of self-monitoring using wearable activity trackers (e.g., Samsung Galaxy Gear[58], FitBit[59], smartwatches or bracelets). Such devices utilize, e.g.,

[58] http://www.samsung.com/de/consumer/mobile-device/mobilephones/smartphones/SM-V7000WDADBT, accessed April 16, 2014

[59] http://www.fitbit.com, accessed April 16, 2014

gamification [79, 171, 314, 315] and competitive elements like sharing training data on social networks [157] as motivating factors. However, a survey conducted by Euromonitor[60] found that only 25% of interviewees have already downloaded a fitness app, and less than 6% use it daily. Another study [179] showed that more than half of all U.S. consumers who own an activity tracker do not longer use it; a third of U.S. consumers even stopped using it after six months. This means that long-term motivation is still not generated by the presently included motivational factors.

Klasnja and Pratt [157] argue that users need to experience a feeling of progress towards defined goals. We believe that a key factor to this is individualized feedback. Telling users what they are doing wrong and how they can improve their performance will contribute to keeping people engaged. We present an approach to achieve this through exercising assessment, by means of a novel input modality for gym training exercises in Section 3.3.

This chapter gives answers to the following high-level research questions:

- How can MUSED interaction support users in daily health-related tasks?
- How can MUSED interaction enable novel interaction in the context of personal fitness?

This chapter is partially based on papers we have published between 2011 and 2013 [169, 221, 231, 233].

3.2 Everyday Object Identification

The use case for our first example, situated in the domain of ADL, is taken from the area of physical mobile interaction, i.e., the interaction between mobile devices and physical objects in the sense of Välkynnen *et al.* and Rukzio *et al.* [290, 325].

A prerequisite for physical mobile interaction is that objects can be uniquely identified by the mobile device. To this end, there exist two possible approaches. First, objects can be made active by themselves, e.g., by embedding sensors and communication facilities. This idea is comprised in the terms “Smart Objects”, “Intelligent Objects”, or, in particular, “Cognitive Objects” [170, 232] (relating to the fact that these objects have cognitive abilities to perceive data from the environment). Second, mobile devices can account for the interaction process entirely, with the objects remaining passive. This requires novel software on the side of the mobile devices, but has the advantage that non-augmented, everyday object can be used. In this chapter, we focus on the second approach.

3.2.1 Use Case: MobiMed – Mobile Medication Package Recognition

As a concrete use case, we present MobiMed, a mobile medication package identifier. MobiMed can be imagined as a digital package insert replacement, providing detailed information on a drug’s active ingredients, application areas, intake instructions, or side effects. It can be applied, e.g., when the original package insert is lost, too small to read, or for

[60] http://blog.euromonitor.com/2013/08/analyst-pulse-the-growing-use-of-mobile-health-and-fitness-apps.html, accessed June 3, 2014

conveniently comparing different medications. In times of food supplements and vitamin compounds, managing multiple drugs is an issue people are struggling with [120]. The aging society aggravates this problem, as multimorbidity, and consequently the need to intake different pills, is often an issue for elderly people. While we observe an increase of health-related apps and services, such as pill reminders or drug reference guides, elderly people often have limited skills with technical systems. Novel object interaction techniques could be a bridge for untrained users to use such services.

The following scenario illustrates how MobiMed could serve users in different contexts.

> John, 55, architect, is an active golfer and likes cycling tours in his holidays. He regularly takes food supplements (carotenes, vitamin E) and anticoagulants, since he is a cardiac patient. He needs to take in up to four different medications a day in different intervals, why he is sometimes not sure about the correct dosage. Since he had trouble pulling the blister packages out of the box, he removed the package inserts and cannot refer to the instructions. John uses MobiMed to point at the medication package. The system identifies the drug by the appearance of the box. He gets detailed information on the product and scrolls to the correct dosage instruction.
>
> The other day, John wants to get an influenza medication at the pharmacy. Since he is allergic to acetaminophen, he scans the package with MobiMed and checks whether the product contains the critical substance in order to know if he can safely buy it.

3.2.2 Employed Physical Mobile Interaction Methods

We implemented four interaction methods or modalities, following paradigms of earlier research [290, 325]: touching, scanning, pointing, and text search. Figure 3.1 shows users while performing the different identification methods. While the first three are real *physical* mobile interaction methods, the latter can be seen as a conventional interaction method that will be used as baseline in the comparative study.

Touching

Touching is a proximity-based approach that allows to identify an object by bringing the phone close or directly in contact to it. The objects therefore need to be augmented with electronic tags, e.g., based on radio communication [338], such as RFID or NFC (see Section 2.1.2 for a description of these techniques). NFC-capable smartphones can read the tags from a distance of few centimeters.

To support this method, we enhanced medication packages with NFC tags on which we stored the same unique id as contained in the product bar code. Each drug package held a unique 7-digit (at the time of our experiment, meanwhile 8-digit) number (PZN, Pharmazentralnummer[61]) which is encoded in the bar code and can hence be used to unambiguously recognize a drug.

[61] `http://www.pzn8.de`, accessed June 1, 2014

We used MIFARE Ultralight tags operating at 13.56 MHz (NFC Forum Type 2) with 48 bytes of memory, complying with the ISO 14443A standard. Touching a drug package with a NFC-capable phone reads the PZN stored in the tag and shows the drug's detailed information on the phone.

Scanning

Scanning is a proximity-based approach where the user points at a visual tag with the camera, such as a bar code or a QR (Quick Response) code, according to the definition of O'Neill *et al.* [253]. This interaction method works in close proximity (as far as the device's camera can recognize the tag), but does not require direct proximity as the touching technique.

According to alternative definitions [290, 325], scanning can also denote searching for available wireless services in an environment, such as Bluetooth and WLAN. In our work, however, we use the term in the sense of targeting a visual marker.

For our medication package scenario, we used the scanning interaction mechanism for recognizing packages by their bar code.

After the code has been scanned and recognized, the name of the medication appears in a popup and the user is asked whether it is the searched one. After confirmation, the detail page is displayed. Recognition starts immediately with the camera preview screen, it is not necessary to explicitly take a photo.

Pointing

Pointing denotes the process of recognizing an object by aiming at the object with the smartphone. It is probably the most natural technique, since humans are used to point with their fingers at objects as well [290]. We implement the pointing technique using CBIR, which is based on visual feature recognition. Each drug package has inherent visual features, such as logos, imagery, colors, the shape of the package, and imprinted text. These characteristics can serve for distinguishing packages. We use MSER (Maximally Stable Extremal Regions) [206] as features and match them with a reference database of more than 100,000 images. The algorithm returns a candidate list of potential matches, from which the user selects the desired medication. In our tests, the correct hit used to be almost always among the ten first-ranked results (i.e., the first result page), indicating a satisfying recognition accuracy.

Text Search

In this modality, a *full text* search is performed on all database fields, so that drugs can be found by the PZN as well as by their name, ingredients, side effects, etc. A list of search results is presented to the user, from which she can choose one to see the details.

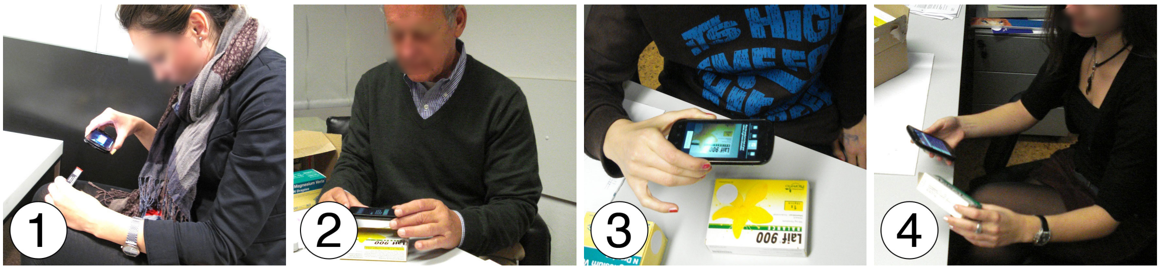


Figure 3.1: The four investigated interaction modalities with MobiMed: scanning (1), touching (2), pointing (3) and text search (4). The images show participants of the laboratory study. Faces were blurred for reasons of privacy.

What about Speech Recognition?

While speech input gained in popularity in the past years, we decided to omit speech as modality in this comparison for several reasons. First, speech input is not a type of direct interaction with the physical object, which we aim to investigate. Second, medication names are often long, contain foreign words, are brand names, or sound similar (especially when they are pronounced incorrectly). This does not only make them difficult to enunciate, but also poses problems to speech recognition engines which are often trained only to standard vocabulary. Third, there often exist variants of medications with different dosages, having identical names, which entails a high risk of inappropriate results. For these reasons, we considered speech input as not appropriate for identifying drugs.

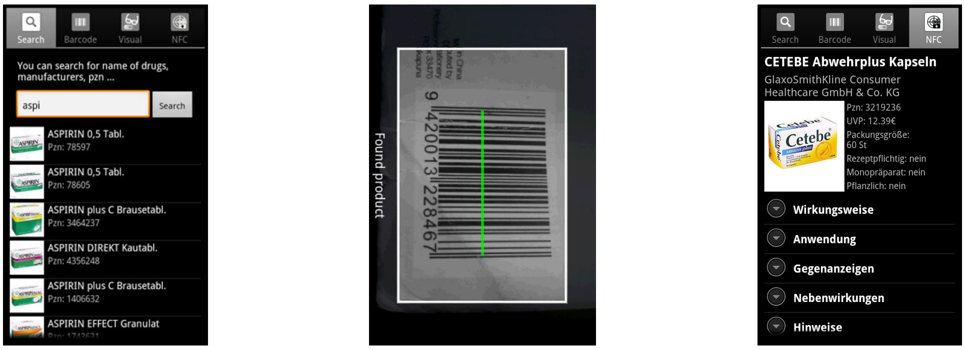
Prototype Implementation

We created a prototype in Android 2.3, which implements the four physical mobile interaction methods described above. The user can select the desired interaction method with tabs at the upper border of the screen (see Figure 3.2). For the interaction method *Touching*, the default NFC API was used. *Scanning* was realized using the ZXing[62] Barcode reader library. For *Pointing*, we used a server-side image recognition engine, similar to [301]. Query images are transferred from the mobile device to the server, where a similarity search is performed and a list of potential matches is returned. The drug information database is likewise located on the server. Once a drug is recognized using one of the four techniques, the database entry is retrieved from the server and displayed on the mobile device. We acknowledge the limitation that the present approach currently requires a network connection. Figure 3.3 shows the interplay between drug package, smartphone and server.

3.2.3 Online Study: User Preferences for Identification Techniques

The first step of the evaluation was to gain large-scale feedback on the different modalities. To that end, an online study was conducted.

[62] ZXing. https://github.com/zxing/zxing, accessed May 19, 2014



You can search for name of drugs, manufacturers, pzn ...

aspi

ASPIRIN 0,5 Tabl.
Pzn: 78597

ASPIRIN 0,5 Tabl.
Pzn: 78605

ASPIRIN plus C Brausetabl.
Pzn: 3464237

ASPIRIN DIREKT Kautabl.
Pzn: 4356248

ASPIRIN plus C Brausetabl.
Pzn: 1406632

ASPIRIN EFFECT Granulat

CETEBE Abwehrplus Kapseln
GlaxoSmithKline Consumer Healthcare GmbH & Co. KG

Pzn: 3219236
UVP: 12.39€
Packungsgröße: 60 St
Rezeptpflichtig: nein
Monopräparat: nein
Pflanzlich: nein

(a) Search screen (b) *Scanning* modality (c) Detail screen

Figure 3.2: User interface of the MobiMed application

Research Questions

We formulated the following research questions (RQ) with relevance to the different interaction techniques:

RQ1(a) What advantages and disadvantages of interaction modalities, as presented in MobiMed, do people see?

RQ2(a) Which method is preferred by users a priori?

RQ3(a) What potential do people see for the MobiMed use case?

Later in this chapter, we will also present a laboratory study, following an iterative research approach. While the online study provides us large-scale qualitative feedback, the goal of the laboratory study is particularly quantitative data. For the laboratory study, we formulate similar research questions RQ1–3. To help relate them with each other, we use the suffix (a) for the online study, and the suffix (b) for the laboratory study.

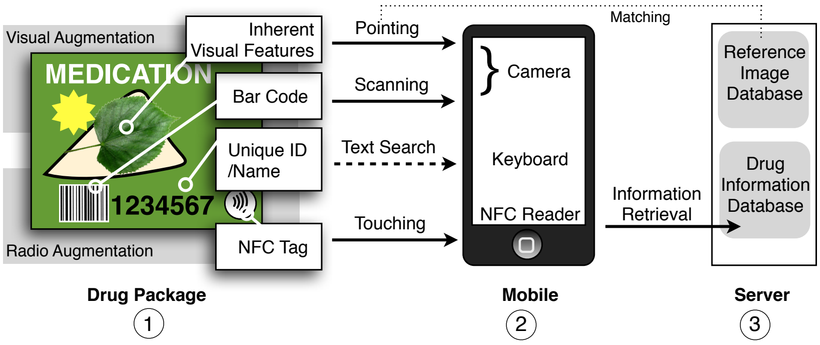


Figure 3.3: Interplay of MobiMed app and MobiMed server in the course of identifying a medication package

Method

The survey was conducted as online questionnaire. First, the concept of MobiMed was introduced. The four interaction modalities were explained detailed in text form and supported by illustrative screenshots of the application (see Figure 3.2 for examples). Thereby we tried to give participants the best possible impression of MobiMed. The questionnaire was carefully formulated to exclude a confirmation bias. For example, we explicitly asked for possible advantages *and* disadvantages of the respective approaches.

Participants were recruited via the Amazon Mechanical Turk platform[63]. The validity of such a crowdsourcing approach is discussed and confirmed in Section 7.5.1. The completion of a questionnaire was compensated with $0.30. An additional reward of $0.50 could be earned for length and quality, in order to motivate subjects to produce high-quality answers.

149 people took part in the study (74 females, 75 males). They were aged between 17 and 79 (M = 31[64], SD = 11). 100 of them lived in the United States. 110 of the participants owned a smartphone.

Results (and Discussion)

RQ1: Advantages/Disadvantages of Identification Techniques

We collected a variety of statements on the presented modalities from free text answer fields, giving an impression what people did and did not like. Hence, results reflect main tendencies and opinions, but cannot expose the spectrum of answers in its entirety.

Scanning Scanning was attributed to be *"quick and convenient"*, *"precise"*, *"easy"*, *"specific"* and *"cool"*. Respondents particularly liked that *"you can know exactly that it is the right product"*, since the bar code uniquely identifies it, even if brand names or packages look similar. There was much familiarity with this technique due to market penetration of bar code scanner apps. Some people reported to *"scan products all the time"* with their phones when shopping for price comparisons. This a priori knowledge might have biased participants' responses towards *scanning* in this study.

As drawbacks, people mentioned the necessity for a bar code and the time to find it on the product. A number of recognition problems were mentioned (due to a too small barcode, bad lighting conditions, a damaged package, or a weak camera). Statements like *"doesn't always work for me" and "sometimes hard to focus the bar code"* indicate that users have experienced these problems themselves.

Touching For NFC-based interaction, respondents highlighted that touching the object with the phone is the only necessary user interaction, which makes the method *"hassle-free"*, *"foolproof"* and suitable for *"old people and non-expert persons"*. They affirmed this technique to allow fast and precise identification, combined with good usability. Other adjectives used were *"modern"*, *"cool"*, and *"satisfying to [...] get so much information so quickly"*.

[63] Mechanical Turk. `https://www.mturk.com`, accessed April 16, 2014

[64] In questionnaires, we ask for the participants' age, not their date of birth. We indicate mean and standard deviations without decimal places to convey the measured accuracy in the magnitude of "years".

Downsides mentioned were the *"extensive"* requirements: a NFC-capable phone, augmented medication packages, and an increased energy consumption on the phone. People hypothesized that NFC, being a novel technology, might be error-prone, which indicates that they were less familiar with NFC and thus more skeptical, compared to bar codes. Other disadvantages addressed costs (NFC augmentation of products would raise their prices and privacy concerns due to the radio technology. As of 2014, mass market prices of RFID tags range between $0.07 and $0.15 per piece, and are expected to drop below $0.05 in near future[65]. Regarding privacy, people seemed to overestimate the proximity range of NFC (which is actually only few centimeters), since also *"interferences with other packages around"* were listed as potential problem.

Pointing Respondents imagined the *pointing* method using feature recognition to be simple, convenient and easy to use. In accordance with earlier findings [325], pointing at objects was considered as very intuitive. One person said that it is *"the most human form of scanning"*. Subjects appreciated that no search for tags or codes is necessary with visual recognition: *"No need to fumble about looking for the bar code on the product"*. Instead, you *"could scan from any angle"*. People came up with the idea that also images from websites could be identified; it would not be necessary to hold the product in their hands. Several mentions pointed out that damaged packages could be recognized as well. A person added that it would be *"excellent [...] for persons with sight or motor skill disabilities"*.

On the negative side, subjects suspected a high processing demand and potentially slower recognition, compared to the other methods. Further, it was noted that visual search, unlike explicit tags, does not provide unambiguous results. Subjects supposed that medications could be confused *"by slight deviations from standard packaging, e.g., a pack with a sticker saying 2 for 1"*, or *"if companies produce packaging designs that are really similar"*. However, one person said: *"Although this may not get your specific product, it can identify similar products. That's incredibly helpful."* Hence, the inherent ambiguity was seen as a strength when searching for similar products, such as different package sizes of the same brand, other dosage forms (powder instead of pills), etc. In our implementation, we make this ambiguity explicit by presenting a list of results, and give the responsibility for choosing the correct one to the user.

Text Search Text search was attested the highest familiarity and respondents liked its inherent accurateness and its multi-functionality. For example, text search allows to search for other keywords than the product name. One person accentuated that *"general terms"* could be used for search, and a *"broad range of results"* would show up. Another said that you can *"find products of the same category, and [...] make a comparison among them"* (even though this was not the goal of the app). In particular, when not knowing what drug they are looking for exactly, subjects considered text search as a good method. An important point mentioned was that the method is independent of sensors and drug package (*"you don't need to be near the product"*), as long as the name is known. This was considered as advantage not only in case of recognition failures, but also in light of the fact that in some countries (including the United States), pills are often handed to patients without original packaging. Respondents also came up with other usage scenarios, e.g.: *"Say you're allergic to acetaminophen you can see what drugs contain it to know what to avoid"*. While this goes beyond the original task

[65] `http://www.rfidjournal.com/faq/show?85`, accessed May 19, 2014

of identifying drugs, they are nevertheless an interesting example of what MobiMed could additionally be used for.

Mentioned drawbacks were the difficulty in relation to the other modalities (e.g., the necessity for on-screen typing, longer search time and potential misspelling, which is likely for complicated drug names and medical terms). Subjects saw the problem of not getting any results due to typos and that *"incorrect wording [...] could end up giving people information on the wrong medicine"*.

RQ2: Preferences for Modalities

Asked for their favorite modality, participants showed a clear preference for *scanning* (48.3%), followed by the other modalities *text search* (25.3%), *touching* (13.6%), and *pointing* (12.2%).

The strong preference for scanning and text search can probably be explained by their high level of familiarity. Most smartphone users are experienced with text search and bar code scanning, while they are less familiar with NFC and visual feature recognition. One person stated: *"I have experience using other software using barcodes, and have liked their ease"*. A respondent who chose text search said: *"This is a tried and true way of researching information"*. Thus, previous knowledge and positive experience may have attracted respondents to choose a familiar method as their favorite in this online study.

It seemed more difficult for the participants to evaluate pointing (visual search) and touching (NFC) without hands-on experience. In particular, recognizing objects just by pointing was partly seen as "science fiction". For NFC, the recognition range was overestimated, which lead to the assumption that closely placed products would interfere with each other and make targeting the desired one difficult. Some subjects even worried about being inundated with information when passing by the shelves in a store without having a hand in the matter.

RQ3: Potential of MobiMed

The question *"Would you use such a system as described above?"* was answered with "yes" by 81.8% of respondents. They liked the idea of finding drug information fast and easily, and envisaged various target groups that could benefit from MobiMed, such as pharmacists, doctors, or people who take multiple drugs. A person said it was "perfect for if you have something at home that you want to find somewhere so you can pick more up or learn more about it".

In average, interviewees would spend \$8.40 for MobiMed (with a standard deviation of \$17.12). The high variance is rooted in the difference between older and younger respondents: Those under 25 would averagely pay \$6.34, those older than 25 in average \$14.01. There are two possible reasons: First, older people might have a higher need for medical applications, so that they see a higher personal value. Second, they might have a higher average income or simply a different attitude towards software pricing. By contrast, younger people are used to get software in mobile app stores for small amounts of money.

3.2.4 Laboratory Study: Efficiency and Usability

Following an iterative approach, we verified our findings of the online study with a smaller number of participants in a hands-on study. Interacting with a prototype allows both quantitative measurements and a more informed judgment on the individual interaction modalities.

Research Questions

We investigated the following research questions:

RQ1(b) Which object identification modality is superior in terms of efficiency?

RQ2(b) Which method is preferred by users in practical use of the MobiMed application?

RQ3(b) What usability and potential do people see for MobiMed *after having used the application* (independent of modalities)?

Note that the research questions correlate with RQ1(a)–3(a) of the online study. However, while the online survey revealed *a priori* findings (i.e., before subjects actually used the application) of a large user group, in the lab study research questions were answered based on experiments and questionnaires *after* subjects had used the application and interaction modalities. Furthermore, RQ1(b) now addresses a quantitative comparison, while RQ1(a) investigated qualitative differences.

RQ1 was evaluated in an experiment, RQ2 and RQ3 with a questionnaire after the experiment.

Method

We conducted a repeated-measure, within-subjects experiment with four conditions. The order of conditions was counterbalanced using a Latin square design [52].

Task

In each condition, subjects had to identify medication packages using a different modality (*touching*, *pointing*, *scanning*, or *text search*). We placed 13 medication packages in a box, out of which subjects had to identify 10 with each modality. Participants were asked to fetch one package at a time out of the box (blind draws; the order was randomized) and to identify it with MobiMed. After successful identification of 10 packages, they were put back in the box and the condition was changed. In the *text search* condition, users were free to either type drug name or identification number (PZN) which was printed below the bar code on each package. The packages were augmented with NFC tags to work with the *touching* modality.

The experiment was conducted with a Samsung Nexus S phone. Prior to the experiment, subjects were allowed to make themselves familiar with the phone and the MobiMed application. The experiments were conducted at a table in a separated, brightly lit room at a medical office. During the experiment, subjects were encouraged to express any thoughts that came to their mind (“think aloud” method [328]). The experiment took about 30 minutes per participant. A researcher was present during the entire time.

Data Collection

For RQ1, we measured the task time (efficiency), which we define as the time interval between the beginning of the interaction and the appearance of the result screen on the smartphone. For each modality, subjects pressed a button to start the recognition, which started a timer. The timer was stopped upon appearance of the drug's description page. Thus, for touching and scanning, task time was equivalent to the recognition time of the NFC tag or the bar code. For pointing and text search, a result list was shown first, since these methods can return ambiguous hits. In these cases, the task time consisted of the recognition time plus the selection time of the correct list item. The timer was always stopped upon the first appearance of the drug's description screen, i.e., it was assumed that no corrections of the user's choice from the list were required.

For RQ2 (preference), we asked subjects how they liked each modality and whether they would use it in the future. For RQ3 (usability), we used the System Usability Score (SUS) [34] which consists of a set of 10 Likert items (1 = strongly disagree, 5 = strongly agree).

Participants

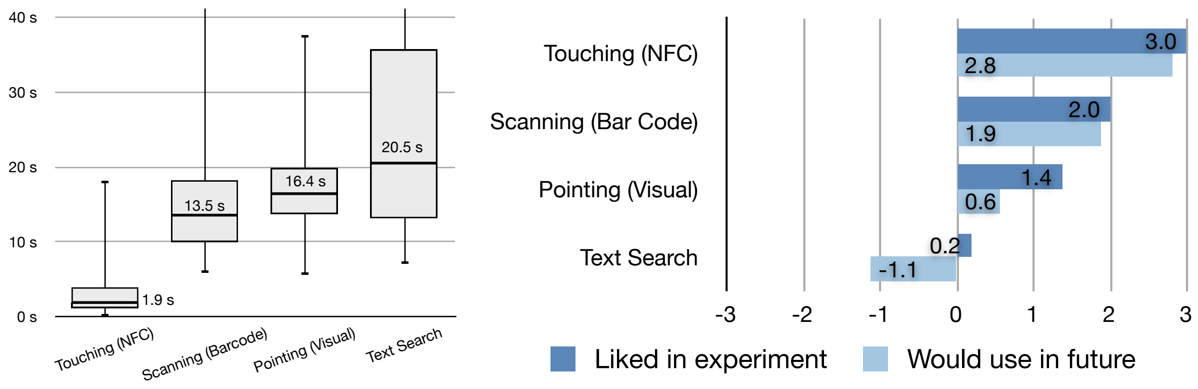
16 people (6 females, 10 males), aged between 22 and 69 years (M = 31, SD = 12) took part in the evaluation. All of them had a mobile phone and used it regularly; 9 owned a smartphone. No subject had physical disabilities that could have hindered the execution of the demanded tasks (such as difficulties with holding the smartphone steadily). Participants were recruited among acquaintances of some of the researchers; none were involved in the project. Subjects did not receive monetary compensation for their participation.

Results (and Discussion)

RQ1: Efficiency

With one-way repeated-measure ANOVA [205], we found a significant effect of the modality on task time ($F(3, 45) = 91.21$, $p < 0.0001$, partial $\eta^2 = 0.99$). Subjects identified a drug in averagely 1.8 s with the modality *touching* (SD = 3.7 s). This was significantly faster than *scanning* (13.5 s, SD = 9.6 s), *pointing* (16.4 s, SD = 6.1 s) and *text search* (20.5 s, SD = 22.9 s). The measurements are visualized in Figure 3.4a.

The largest standard deviations were observed for *text search* and *scanning*. The reason why subjects struggled with *scanning* was presumably due to the problem that the focusing the camera on small bar codes was difficult. The variance for *text search* reflects the diverging typing capabilities of participants. With a maximum of 102.3 s, text search took more than five times longer than the longest NFC identification (18.1 s). It could also be explained by the length of some drug names, providing no upper bound for text input length. It is worth to mention that these results were still obtained under "ideal" conditions, i.e. under the assumption that subjects selected the correct item from the result list. In practice, the need to correct accidental choices might entail even longer total times for the *text search* modality. This might also be true for camera focus and image capture in non-optimal lighting conditions.



(a) Lab study: Task time

(b) Lab study: User preferences

Figure 3.4: Left: Box plots indicating mean task times for object identification with MobiMed. Boxes represent the interquartile range, whiskers represent extrema. Maxima for scanning (61.7 s) and text search (102.3 s) were cut off for better readability. Right: Likert scale averages on whether users liked a specific identification method and whether they would use this method in the future (-3 = strongly disagree, 3 = strongly agree).

RQ2: User Preferences

Users' preferences for modalities match task times, in that faster modalities were also rated better (see Figure 3.4b).

On a 7-point Likert scale[66] (-3 = strongly disagree, 3 = strongly agree), participants responded with 3.0 that they liked *touching* (SD = 0.0). *Scanning* was rated with 2.0 (SD = 1.2). *Pointing* received a rating of 1.4 (SD = 1.9) and *text search* was rated averagely with 0.2 (SD = 2.0). When asked whether users would use the methods *in future*, the order was the same, but at a lower level of agreement: Subjects agreed with 2.8 (SD = 0.4) that they would like to use the touching method. The scores for scanning and pointing were 1.9 (SD = 1.0) and 0.6 (SD = 2.0), respectively. The wish to use text search was expressed with -1.1 (SD = 1.8).

RQ3: Usability and Acceptance of MobiMed

MobiMed received a SUS score of 88.0 points out of 100 possible points. According to Bangor *et al.* [15], SUS scores above 85.5 are considered as "excellent". Hence, we can assume that interaction with MobiMed does not entail major usability problems. Figure 3.5 illustrates the SUS values for the individual items. Positive items were usually rated with a score of more than 4, except the statement *"I think that I would like to use this system frequently"*, which was rated with 3.3. Subjects averagely agreed with unfavorable items with a score of less than 1.4. Only one participant stated to *"need support of a technical person"* to use the system.

Acceptance and Comments In order to learn more on MobiMed's utility in everyday life, we asked people by which information sources they usually inform themselves about medications, their active ingredients, dosage, and side effects. 75% ask their doctor or pharmacist, 69% read the package insert. 56% stated to consult books or the internet, 13% use other

[66] As the response format approximates an interval-level measurement, we use mean values to report Likert responses in this dissertation.

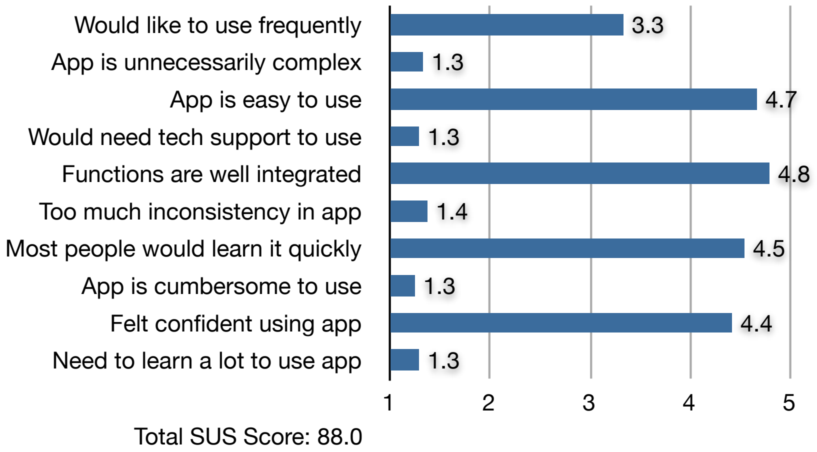


Figure 3.5: Individual aspects of the System Usability Score (SUS) for MobiMed on a 5-point Likert scale (1 = lowest, 5 =highest agreement with statements). Statements are abbreviated; for exact wording, see [34].

sources. 75% of subjects use more than one single source to get information on drugs. After the study, 14 out of 16 participants (88%) declared that they were interested to use MobiMed as alternative source to inform themselves on drugs. Not only is this an indicator that subjects appreciated the prototype but MobiMed was also the most popular individual source of information of all other ones among the participants in this study.

Asked for desired additional features in MobiMed, subjects came up first and foremost with shopping-related functions: price lookup, finding cheaper generic products, providing a list of suppliers, and the possibility of direct order (except drugs that are only available by prescription). They were also interested in active ingredient analysis: MobiMed could suggest products that show fewer cross-correlated side effects for a specific combination of components. One participant suggested a tool that helps diagnosing based on symptoms a user enters. Several subjects were interested in a personalized medication management tool that allows to manage drug intake, creates medication lists and reminds users of their pill intake.

3.2.5 Discussion and Lessons Learned

Having investigated advantages and disadvantages, efficiency and user acceptance of four object identification modalities at the example of MobiMed, we summarize our findings and observations in five main points. Results from both studies coincide in large parts, although we found some divergences, which we also try to explain below.

Physical Mobile Interaction is Popular and Efficient In the lab study, subjects preferred the modalities involving physical mobile interaction (touching, pointing, scanning) over conventional text search, and would also prefer them in future use. Physical mobile interaction modalities were faster for identifying drug packages than text search, and the standard deviations were significantly lower. These findings motivate a consideration of alternative modalities for mobile interaction, and especially physical interaction with real-world objects, making the interaction with the physical world more effective and intuitive. While the capabilities of mobile devices were not sufficient few years ago to support, e.g., visual feature recognition as employed in the *pointing* technique, they are now not a limiting factor any more.

Efficiency is an Important Criterion The preferences for the individual modalities correlate with the measured task completion times (faster methods were ranked higher), which proves that efficiency is an important criterion in the eyes of the users. *Touching* and *scanning* were the fastest and most popular methods. *Touching* was significantly faster than scanning, and also adopted more positively by subjects in the lab study.

Trying is Crucial In the online survey subjects answered differently: Almost half of them stated that they liked *scanning* most, followed by *text search*. Here, subjects seemed to favor modalities they already knew. While scanning and text search were familiar modalities, they could imagine worse how touching using NFC or pointing using visual search would work. An important "lesson learned" is here that when proposing a new interaction method, people should have the possibility to try it.

Technological Change Enables Novel Interaction Modalities The processing power of state-of-the-art mobile phones enables novel techniques like the feature-based recognition modality (*pointing*). Such novel interaction methods can have a significant impact. As for pointing, the fact that no marker augmentation is required opens up this method for a variety of applications beyond the scenario we investigated in the presented experiment. In the online survey, this potential was already recognized by subjects, mentioning that there is no need to search for the bar code and that it is the most natural way of physical mobile interaction.

In our study, pointing was still not as popular and not as fast as touching and scanning. However, it almost reached the performance of *scanning*, and had no outliers like *scanning* due to unreadable bar codes. Improved implementations and the continuous rise of processing power and memory in mobile devices will further increase recognition speed and reliability of visual search. Most importantly, what we learn here is that we should see the technological progress as a chance to rethink which interaction methods are possible.

Modalities Must Be Chosen With Scenario Respondents identified various advantages and disadvantages of the individual modalities. It became clear that the "best" interaction method depends on the selected scenario. One example is the question whether a method should return unambiguous results or a result list. For product comparisons, multiple results as produced by visual search are desired, while reliable information for drug intake at home requires a method that provides a unique result, e.g., scanning the bar code. Visual search can be an interesting alternative when the bar code is not readable or invisible, e.g., when drug packages are placed behind glass in the pharmacy, or information should be retrieved from a picture, e.g., on a website. What we learn is that the matching between scenario and used modality must be carefully made, and that it might be a good idea to offer different ways to perform a task. It also turned out that *performance* is only one factor for user preferences, which suggests future research on factors that influence the likability of physical mobile interaction techniques for a specific scenario.

High Acceptance of the Medical Use Case Responses of both the online survey and the lab study showed a high level of interest in medical apps, and in MobiMed in particular. This might be related to an increased awareness for a healthy lifestyle (which includes interest in food supplements and ingredients), but also to a rising need for medication support in light of the aging society. MobiMed was evaluated by subjects as helpful complement to other information sources, such as the pharmacist's advice. Users are less intimidated to consult a digital app rather than to ask a doctor. Thus, especially for elderly people who have problems with reading the (original) package insert, the adjustable font sizes of a mobile application could provide a benefit. Thinking one step further, apps like MobiMed could provide individual content (e.g., age-appropriate, easier to understand than the original package insert).

3.3 Automatic Physical Exercise Assessment

With our second example from the area of health and fitness, we look at smartphone-supported personalized fitness training. Our choice of this application area is motivated by the fact that regular physical exercising is crucial for a healthy lifestyle. Opposed to that, people's knowledge about safe and effective exercising is in many cases not sufficient. Moreover, constant exercising requires a high level of motivation. As motivated in the beginning of this chapter in Section 3.1, we see individualized feedback as a potential solution to this problem.

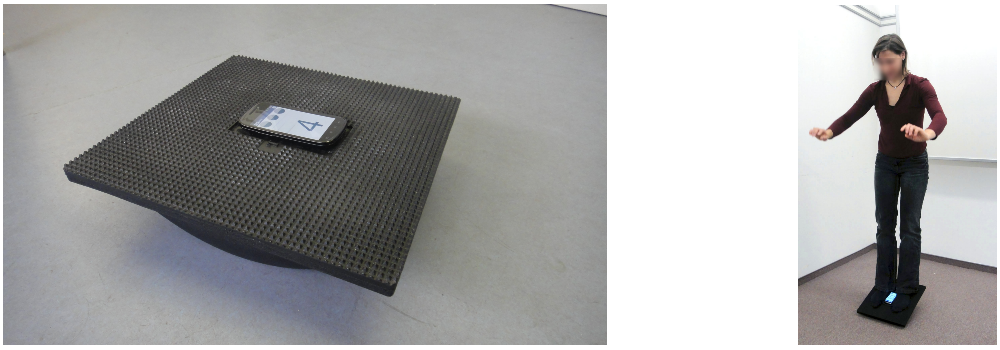
We propose the use of smartphones as digital personal trainers for physical exercising. Being our daily companions, they are ideal for monitoring and supporting regular physical exercising. What if they additionally could assess the exercise performance and give feedback on how to improve? With their integrated sensing capabilities, they can supervise the exercising performance and track an exercising person's improvement. Thereby, they become the conceptual equivalent of a personal trainer. Compared to their human counterpart, digital personal trainers have the advantages of ubiquitous and permanent availability alongside with negligible costs. Furthermore, privacy and dignity is better preserved by allowing for exercising without permanent supervision of an unknown human fitness trainer and in a nicer environment than a public gym. Arguably, automated skill assessment and individualized feedback also increase and maintain motivation [231], which is crucial for effectiveness, since training needs to be done regularly.

3.3.1 Use Case: GymSkill – A Multimodal, Personalized Fitness Coach

As use case, we developed GymSkill, a smartphone application for ubiquitous monitoring and assessment of balance board exercises. The application we describe in the following can be downloaded from Google Play[67]. The choice for balance board training was made for our explanatory purpose, as this type of exercises is relatively easy to understand, and thereby well-suited as conceptual example for automated motion analysis and skill assessment. The idea of performance tracking can however be transferred to other, more complex sports and sequences of movements (which will then possibly include a larger number of body-worn sensors). In our case, no equipment besides a balance board as depicted in Figure 3.6a[68] and a smartphone is required, supporting our idea of ubiquitous training aid, be it at home, in the

[67] https://play.google.com/store/apps/details?id=de.tum.ei.vmi.fit, accessed June 3, 2014

[68] see, e.g., http://www.thera-band.com/store/products.php?ProductID=17, accessed June 3, 2014



(a) A smartphone running the GymSkill application attached to a balance board.

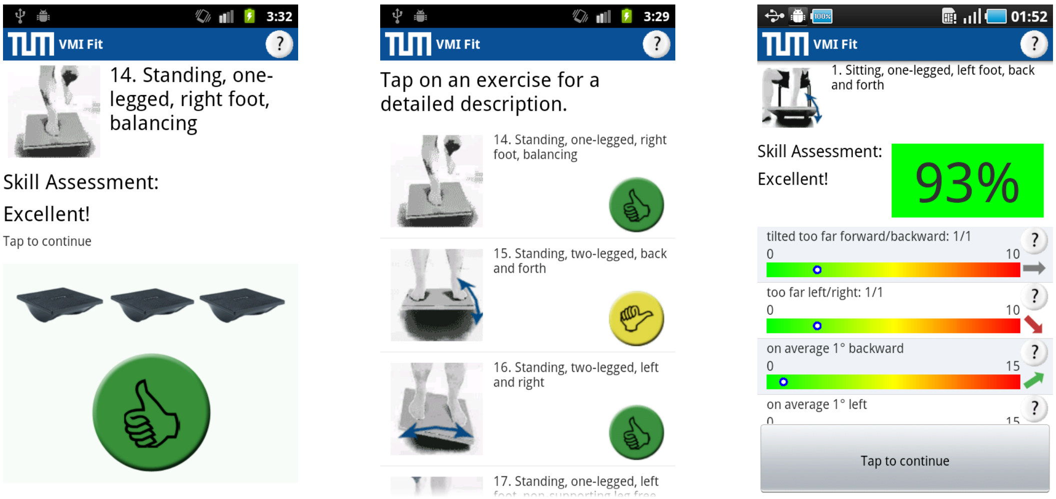
(b) A person performing a balance board exercise supported by GymSkill

Figure 3.6: The smartphone's accelerometer and magnetometer capture the training device's motion as a basis for GymSkill's exercise skill assessment.

gym, inside or outside. Exercises that can be performed with balance boards address a wide target group of all age classes [167]. Balance board exercise programs train, e.g., ankles, the equilibrium sense, and contribute to the overall fitness. Of special interest in this chapter is the (multimodal) interaction between the GymSkill application, the user, and the training device.

GymSkill records and analyzes the performance of balance board exercises with relation to different parameters. On that basis, the application is not only able to assess the exercise quality, but also to provide details on how the exercising person can improve. This targeted, individual feedback can point to problem areas or help identify exercises that need particular improvement.

The GymSkill application is an example for multimodal interaction in several ways. During the training, the smartphone with the GymSkill application running is attached to the balance board. That way, the device can record all movements of the board, which are in turn performed by the human user. The motion as input modality is mediated by the physical device used. This is a form of implicit interaction (see Section 2.1.1). Motion here is interpreted as input modality, feeding the application with training data while performing the exercise. To further simplify the setup of GymSkill, we implemented the *touching* modality as presented in a previous chapter of this dissertation (see Chapter 3.2) using NFC. Different training devices with different levels of difficulty, e.g., rocker boards with one degree of freedom (DOF), wobble boards with two DOF, or boards of different manufacturers, require a re-calibration of the application, as the physical properties of these boards differ between each other. With the *touching* modality, i.e., by simply placing the smartphone on a (NFC-augmented) balance board, these calibration settings are made automatically. This is realized by IDs stored in the NFC tags that are associated with manufacturer- and device-specific calibration data on the phone. Moreover, the tag recognition launches the GymSkill application and loads previous exercising programs where they have been interrupted, so that the effort for beginning a training session is minimized. Our intention is to facilitate training in short slots of free time, as people can immediately start exercising. Referring to the dimension of multimodality as outlined in Section 2.1.1, GymSkill is an example for sequential multimodality (temporal dimension).



14. Standing, one-legged, right foot, balancing

Skill Assessment:

Excellent!

Tap to continue

Tap on an exercise for a detailed description.

14. Standing, one-legged, right foot, balancing

15. Standing, two-legged, back and forth

16. Standing, two-legged, left and right

17. Standing, one-legged, left

1. Sitting, one-legged, left foot, back and forth

Skill Assessment: 93%

Excellent!

tilted too far forward/backward: 1/1

too far left/right: 1/1

on average 1° backward

on average 1° left

Tap to continue

(a) Simple skill assessment (b) Exercise list (c) Enhanced assessment

Figure 3.7: User interface of the GymSkill application. (a) Skill assessment after exercising, based on the evaluated sensor information. (b) In the exercise list, exercises that need further training can easily be identified in the first version of the GymSkill prototype. (c) Global score, detailed skill report on individual criteria and trends (small arrows) in the second iteration of GymSkill.

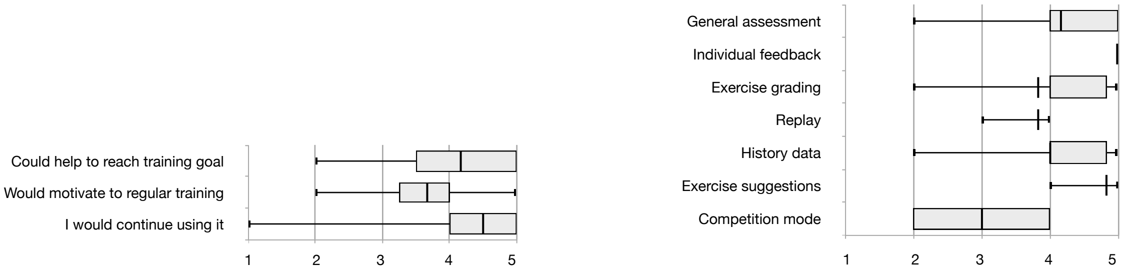
3.3.2 Exercises and Ground Truth Data Acquisition

With the help of a sports medicine specialist, we developed a set of 20 consecutive balance board exercises, comprising tilting and balancing movements in different poses (back and forth, left and right, while sitting or standing, etc.). The exercises were designed such that the level of difficulty was increasing when performing the entire set; but exercises are also adequate for training individual parts of the body (e.g., for rehabilitation after surgery).

In order to gain training data for automatic exercise assessment, six persons (2 females, 4 males) aged from 25 to 33 years (M = 29, SD = 3) performed sessions of 20 balance board exercises twice a day for a period of five days. In total, 1,200 exercise records were captured. Therefore, a smartphone running the GymSkill application was attached to the balance board and accelerometer and magnetometer data was recorded. In addition, the performances were video-recorded. In cooperation with an expert clinician, we created a rating scheme incorporating individual quality aspects such as regularity and movement angles, and used it to assess the video-taped performances with the help of the expert. These assessments were used as ground truth for the evaluation of our algorithm (see Section 3.3.5).

3.3.3 User Feedback

After the participants had trained five consecutive days with GymSkill, they answered a short questionnaire. Since the prototype used for this study was not yet able to provide real-time feedback, the questions did not cover the actual skill assessment, but rather the handling of the application and its potential to motivate regular exercising.



(a) Feedback on prototype after 5 days of training

(b) Most attractive features

Figure 3.8: (a) After 5 days of training with a GymSkill prototype, users believe the app could motivate and help to reach training goals faster. (b) Individual feedback and suggestions of exercises were particularly attractive features. Answers given on a Likert Scale (1 = fully disagree, 5 = fully agree).

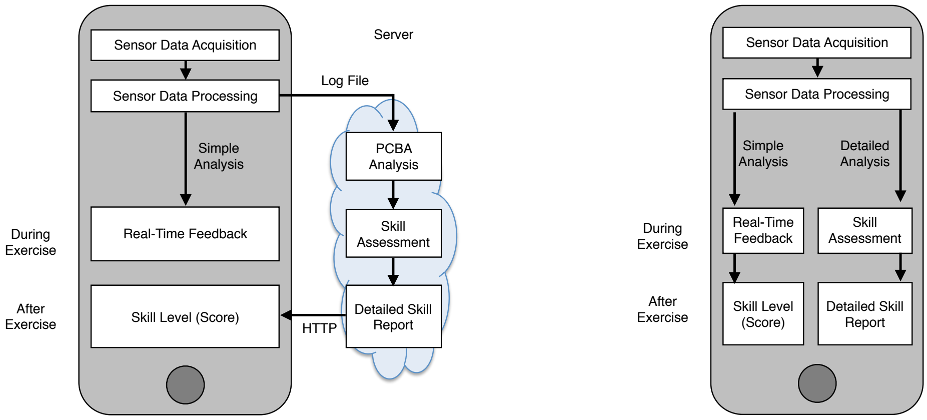
A summary of the results can be seen in Figure 3.8. Subjects stated that GymSkill could help to reach a training goal faster with an average of 4.2 (SD = 1.3) on a 5-point Likert scale (1 = fully disagree, 5 = fully agree). GymSkill's potential to maintain long-term motivation was confirmed with an average of 3.7 (SD = 1.0). Asked for the most attractive potential features of a personal fitness trainer like GymSkill, individual exercise feedback was mentioned with 5.0 (SD = 0.0), followed by individual exercise suggestions (4.8, SD = 0.4). Subjects stated that they would use the application regularly with averagely 3.6 (SD = 1.0). The handling of the system (placing the phone on the board to record data) was evaluated as easy (average agreement: 4.0, SD = 1.1); placing the phone on the board to record data was apparently not perceived as problematic.

3.3.4 First Iteration: Principal Component Breakdown Analysis

The choice of the assessment method was made based on the requirements on the parameters the system should be able to evaluate (quality measures). The list of quality measures was created in cooperation with a sports medicine specialist and used for a manual assessment (as mentioned above). Subsequently, in cooperation with the University of Newcastle [231], an algorithm that builds on and extends principal component analysis (PCA), was developed, which we call Principal Component Breakdown Analysis (PCBA). As we are in this thesis interested in findings on the multimodal interaction with GymSkill, we will only describe the algorithm in its basics. For a more detailed description see our previous GymSkill publications [169, 231].

Quality Measures

The following quality measures cover the most important aspects of the performed motion. The criteria are targeted at balance board exercises; however, some of the criteria can be applied to other exercises that require recurrent movements with quality constraints related to smoothness and efficiency. This entails that the PCBA algorithm is applicable and generalizable for such exercises as well.



(a) Iteration 1: The smartphone records sensor data during exercising, which are processed by a server after the training to generate skill assessment.

(b) Iteration 2: Data is processed on the phone for real-time feedback as well as sophisticated feedback addressing individual aspects after the execution.

Figure 3.9: Iterations of the GymSkill application.

Smoothness and Continuity of Movement For continuous exercises, as they are typical for gym-based training, it is important to maintain smooth motion. In order to remain relatively independent of the particular exercise and to avoid the excessive use of prior knowledge, a novel local assessment approach has been developed (see second iteration in Section 3.3.5).

Global Motion Quality Each exercise requires the user to perform particular motion sequences. The assessment on how well these motions were performed is crucial for the assessment of the quality of the performed task.

Usage of Board's Degrees of Freedom If a task requires the user to fully displace the board along at least one degree of freedom, the fraction to which she uses this opportunity while avoiding extreme postures (e.g., touching the ground) provides another valuable measure for exercise performance.

Algorithm

The goal of the automated assessment is to estimate measures for the aforementioned aspects and to combine them into a single performance metric.

As a basis for all calculations, our algorithm uses the orientation values recorded by the smartphone (azimuth, pitch, roll). The basic idea of the performance analysis is to look for unusual sections of the data, compared to the rest. We assume that sensor data for reoccurring, smooth movement, as it is desired for the type of exercises we look at, should share certain (unknown) statistical properties. Deviations in the data correspond to irregularities (breakdowns), which occur, e.g., when the exercising person hesitates or gets stuck while exercising. This brings us to the name Principal Component *Breakdown* Analysis.

This offline algorithm extracts portions of the data using a sliding window and projects them to a lower-dimensional subspace. Its dimensionality is determined by the analysis of the eigenvalue spectrum. Using the lower-dimensional projection, the original frames are reconstructed with a pre-defined variance threshold (typically 95%). The less regular the original data is, the more eigenvectors the PCA models will require to preserve that amount of variance of the original data. The resulting reconstruction errors are then used as measure for the quality of the underlying movement. By fixating the target dimensionality and implicitly analyzing the modeled variance, an effective quality assessment is gained. Unlike standard techniques for time-series analysis (e.g., [140, 252]) this approach processes sensor data of arbitrary dimensionality. By that means, we avoid flattening the data to one-dimensional sequences, which could lead to loss of potentially important information.

For unsupervised analysis, the "correct" frame length, i.e., the size of the neighborhood that needs to be analyzed for discovering potential characteristic breakdowns, needs to be known. Unfortunately, this information is typically not available for practical applications. To overcome this dilemma, we employ a multi-scale approach by performing quality assessment on a pyramidal adjacency representation of sensor values with increasing frame lengths, which is comparable to the general idea of Wavelet analysis or the approach presented in [252]. This approach focuses on self-similarity and breakdowns with relation to global characteristics and yields pyramidal-shaped visualizations as depicted in Figure 3.10.

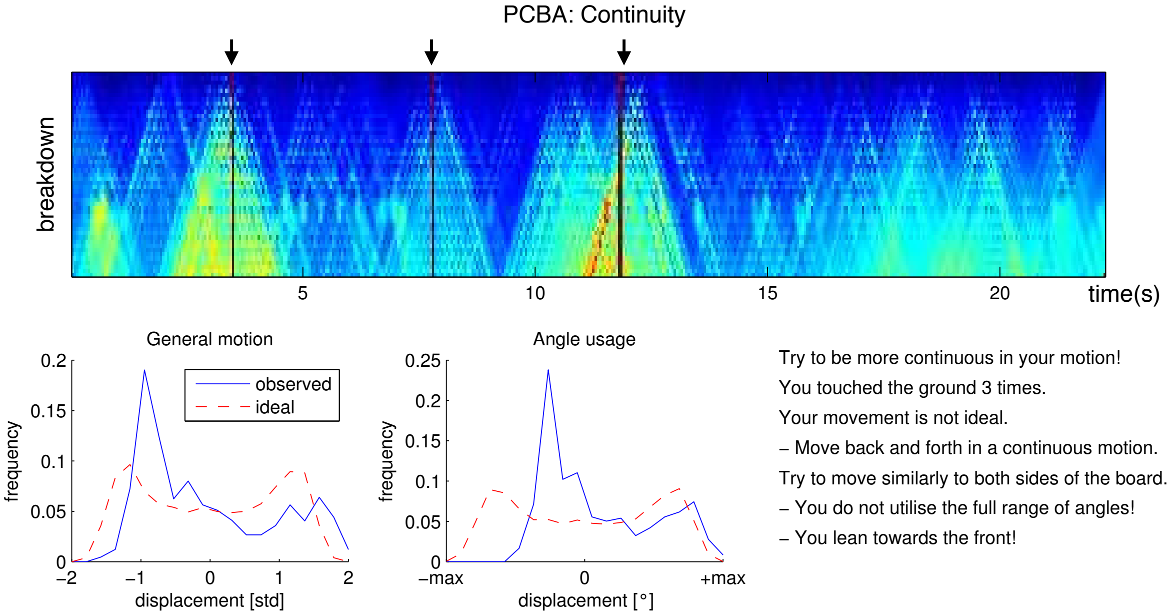
In addition to the local self-similarity analysis, we analyze how much the performed global motion resembles the "optimal" motion (the "gold standard"). The motion axis that provides the dominant signal is estimated and an empirical distribution function is derived. This empirical distribution is compared to an ideal distribution function (see red dotted line in Figure 3.10). The parameters were motivated by insights provided by an expert clinician. Alternatively, the performance of a skilled professional (or the average of multiple such performances) can be used to empirically estimate the ideal behavior.

Implementation

GymSkill is designed as client-server system, consisting of the training app and the server-based analysis component (see Figure 3.9a). The smartphone is attached to the training device, so that all movements of the balance board can be recorded by the GymSkill application.

The logged information is sent to a server, where an offline, retrospective exercise assessment is performed in form of the above-described PCBA analysis. The calculated skill level sent back to the mobile application and visualized as a three-step "thumbs up/thumbs down" visual feedback. In addition, more sophisticated graph visualizations are generated on the server (see Figure 3.10 for one example), which can be used for a later in-depth review. These visualizations are generated with a MATLAB script and show the performance over time and help the user identify specific problems. The server also generates textual feedback from the analysis, supporting the intelligibility for users and providing effective recommendations for further trainings (see Figure 3.10).

Moreover, basic feedback is already given during the performance. The therefore necessary simple analysis methods are performed directly on the mobile device. The remaining number of repetitions during an exercise is displayed, and the tilting angle of the board is graphically



Try to be more continuous in your motion!
You touched the ground 3 times.
Your movement is not ideal.
– Move back and forth in a continuous motion.
Try to move similarly to both sides of the board.
– You do not utilise the full range of angles!
– You lean towards the front!

Figure 3.10: Detailed analysis of a performed exercise. The PCBA highlights differences in continuity of the performed motion (local analysis), and discrepancies to the ideal motion (global analysis). Quality breakdowns over time are highlighted with arrows. The quality assessment does not only result in a global score, but also in detailed insights on how the performance can be improved, which are extracted as textual feedback. Depiction based on [231] (work in cooperation with University of Newcastle).

visualized. Excessive displacement (further than the optimal range) is signaled with red color and with a warning sound.

Usability has played a major role for the development of the GymSkill application. The feedback during training is multimodal (sound and salient visual changes) to be peripherally perceivable and to allow the user focus on the exercising task. The catalog of exercises contains textual descriptions and images for a clear explanation of how an exercise must be performed. The quality assessments of previous trials shown in the exercise list communicate at a glance which exercises need further training (see left and middle image in 3.7).

Evaluation and Discussion

For a qualitative evaluation, we picked a random subset of exercise performances of differently skilled persons (out of our ground truth data, see Section 3.3.2) and analyzed the generated PCBA renderings opposed to their video recordings and (reference) expert assessments. Figure 3.10 shows one example performance of a back-and-forth tilting exercise with the duration of 22 seconds. The colored continuity diagram visualizes the overall smoothness and exhibits three occurrences of breakdowns (yellowish/greenish pyramids highlighted with arrows) in otherwise quite harmonic motion. The coloring of the last few seconds corresponds to the participant stepping off the balance board. Further, the bottom left diagram compares the observed motion to the (dotted red) ideal distribution function (normalized regarding standard deviation) and unveils that the subject does not follow a harmonic motion. The bottom right diagram shows that not the whole degree of freedom available on the board was used. Additionally, the mean of the recorded board positions is displaced from the calibrated zero position, which indicates that the subject's posture is not balanced (in this case, leaning towards the front).

In our qualitative comparison, we found that the generated visualizations correctly communicate the main aspects of the expert's judgements on the particular exercises in the dimensions continuity, general motion quality and angle usage. GymSkill proves suitable for assessing the overall quality of exercises and unveils typical exercising errors like deficient smoothness in movement or not using the available freedom of motion.

3.3.5 Second Iteration: Criteria-Based On-Device Assessment

With the second iteration of GymSkill, we redesigned the previously described system in terms of comprehensibility, location- and time-independent training, and physical interaction with the training device, incorporating the gathered feedback from the users.

First, the skill assessment is now conducted and presented to the user entirely on the mobile phone. This accounts for the fact that motivation is increased when people can analyze their trainings directly afterwards a performance and *in situ*. Previously, viewing the PCBA renderings on a second screen had a higher potential for interrupting the training task. The newly introduced on-device computation and presentation makes GymSkill not only independent of an external server or internet connection, but also from a second screen to review the assessment. This is a step towards ubiquitous exercising and training.

While the PCBA renderings of the previous version provided a good comprehensive overview on a performance at a glance, we strived for a method to more intuitively address individual qualities of performed exercises. Using a new approach for the assessment, sub-scores (see Figure 3.7) make transparent more directly in which aspects the user can improve. That way, we make the feedback more intelligible for average users, while the PCBA renderings required prior knowledge.

In the following, we describe the quality measures and implementation of the on-device skill assessment algorithm incorporated in the second iteration.

Quality Measures

The quality assessment consists of a weighted sum of sub-scores that are assigned for individual quality measures. Perfect motion yields the maximal possible score; for suboptimal behavior, scoring points are subtracted. The quality measures and weights (with slight differences between tilting and balancing exercises) were specified with the help of an expert clinician. In the following, we list the quality measures and briefly describe their implementation. For a more detailed explanation, see [169].

Touching the Ground/Balance The maximal possible score of 10 points is awarded if the user stays within the desired deflection angles and succeeds to avoid extreme postures. This means that the board is not tilted to the maximal possible angle of deflection (known from initial calibration). In that case, it is assumed that the edge of the board has touched the ground, which is undesired. For each occurrence, one point is subtracted.

For balance exercises, a deviation from the neutral position by a threshold of more than ± 5° (user-adjustable) results in subtraction of one point.

Repetition Count/Exercise Length The score in this category is the ratio of actual and required exercise repetitions. In order to count repetitions and assess other cycle-based quality factors, we detect and count zero crossings in the orientation data. Under the assumption that we only look at one movement direction (which is the case for our exercise set), in the ideal case, two zero crossings of the board's orientation make one repetition. In practice, when the user is unable to keep steady on one side, more than two zero crossings can occur in one repetition. This problem was addressed by filtering in the orientation and time domain with experimentally determined values. We consider values in the interval of ±2° deviation from zero as zero crossing. Multiple zero crossings in an interval smaller than three samples (corresponding to 90 ms) were unified to one.

For balance exercises, the ratio of actual and required exercise length is used for this score.

Pace This aspect scores the length of repetitions, whereat a periodic time between 0.9 s and 2 s is considered as optimal, and periodic times shorter than 0.5 s or longer than 4.5 s are scored with 0 points. This sub-score is not used for balance exercises.

Smoothness The smoothness value is calculated from the variance of the length of repetitions and the variance of distortions within repetitions. Distortions are detected by Fourier spectrum analysis. While smooth motion of the balance board resembles a sine function, a high number of components in the frequency domain can indicate tremors and other irregularities. For balance exercises, the smoothness is expressed by the variance of angles during the performance.

Overall Correctness In this sub-score, multiple criteria are bundled, such as the average pitch and roll angle (i.e., whether the user tends to lean forward or backward) and average amplitudes. The features incorporated here correspond to the human understanding of correct exercise execution.

Implementation

The algorithms to determine the above quality measures were implemented in Android as part of the GymSkill application. The updated structure of the application is depicted in Figure 3.9b. In contrast to the PCBA renderings, the feedback is now displayed directly on the smartphone after the completion of each individual exercise. Consequently, the exercise assessment presentation was changed and optimized for small screens (see right screenshot in Figure 3.7). Besides the overall score (in form of a percentage value), the visualization graphically shows the sub-scores, enabling to track down problem areas that need particular attention when repeating the exercise. By small trend arrows, users can identify at a glance which quality aspect has contributed to an improvement of the global assessment.

Evaluation

The accuracy of our skill level scores was evaluated against the expert assessment on the recorded training set in an exercise-by-exercise comparison. As some of the expert's criteria cannot be assessed automatically from the sensor data (e.g., body posture), the expert score was recalculated after excluding these criteria.

A Pearson product-moment correlation coefficient (Pearson's r) was computed to assess the relationship between automatic and expert assessments. We found a positive correlation both for dynamic ($r = 0.51$) and for static exercises ($r = 0.76$). The average difference (bias) between the score calculated by our algorithm and the expert assessment was 3.48 points for dynamic and 0.08 points for static exercises (in relation to a total score of 100 points). This indicates the robustness of our algorithm and the validity of the employed scoring scheme.

The comparisons of the scores are summarized in Table 3.1. For dynamic exercises, the assessment error was smaller than 10 points in 77.86% and smaller than 20 points in 94.27% of the cases. Static exercise assessments showed an error of less than 10 points in 89.58% of the cases, and of less than 15 points in 98.96%. The accuracy differed among the individual criteria: While repetitions were detected very precisely with less than ± 2 points error in more than 96% of all cases, only about 75% of ground touches could be detected. The largest differences were observed in pace; only about 68% of the automatically assessed exercises had an error smaller than 2 points.

It should be mentioned that the evaluation was conducted on magnetometer data, as these provided superior results than accelerometer data (see Table 3.2 as comparison). Since the assessment was conducted using previously recorded data, the magnetometer sensors were uncalibrated. Instead, we used the (calibrated) accelerometer data to calculate the offset to the magnetic field components a posteriori. We also note that the device the experiment was conducted with (a Samsung Nexus One) did not have a gyroscope. We are optimistic that measurements with up-to-date hardware could yield even more satisfying results.

3.3.6 Discussion

We reported on the iterative development process of GymSkill, a system for individualized assessment and feedback on physical exercising through sensor data. GymSkill is a representative example for future mobile interactive applications that incorporate activity recognition, processing and advanced reasoning and interact with physical devices such as exercising equipment. We proved the acceptance of such mobile assistance systems and their potential to motivate users to regular activity in a five day study. Our automated skill assessment proved to be accurate in an evaluation against expert assessments.

In the future, the system could be extended in various ways. More exercise types with more degrees of freedom (circular movements) or with one board for each foot (individual left and right foot movements) can be integrated. Coupling with devices like heart rate monitors, e.g., using ANT+, would further increase the sensed database and allow for further, more detailed physical and physiological assessments. GymSkill could be integrated with health platforms [14, 281] and interact as remote component of a home infrastructure in relation

	bias	$\|\Delta\| < 10$ points	$\|\Delta\| < 20$ points	correlation (Pearson's r)
dynamic	3.48	77.86%	94.27%	0.51
static	0.08	89.58%	98.96%	0.76

Table 3.1: Match of assessment based on our algorithm with expert "reference" score (measured with magnetometer data)

	bias	$\|\Delta\| < 10$ points	$\|\Delta\| < 15$ points	correlation (Pearson's r)
dynamic	7.01	65.89%	90.10%	0.59
static	0.99	90.63%	97.92%	0.78

Table 3.2: Match of assessment based on our algorithm with expert "reference" score (measured with accelerometer data)

bias: average of |automatic score – expert score)|
$|\boldsymbol{\Delta}| < \mathbf{n}$ **points**: percentage of automatic scores this far from expert scores
correlation: Pearson's product-moment correlation coefficient (Pearson's r) [261]

with other health and fitness applications. The smartphone has the advantage of immediate feedback and time- and location-independent training. With our second iteration, we addressed the limitations in feedback presentation on the small phone display by textual, aspect-based feedback.

3.4 Summary and Lessons Learned

In this chapter, we have presented and evaluated approaches for personal fitness and ADL, involving MUSED interaction. We now summarize the advantages of novel interaction methods involving different modalities that we have found in our research.

Usability Improvement

MUSED interaction in the area of physical object identification (Chapter 3.2) showed superior usability compared to the "standard method" (text search). The MUSED modalities *touching*, *pointing* and *scanning* were not only faster and easier (especially with relation to complicated medication names), but also evaluated as more intuitive and more natural by participants. The target group of senior citizens can use these modalities to retrieve medication information even if they would not be able to operate the (small) on-screen keyboard. This is a good example for how MUSED interaction modalities can be door openers for today's and tomorrow's seniors, who have a general understanding of mobile technology, but limited motor or visual abilities. In the health and fitness domain (Chapter 3.3), we have presented with GymSkill a personal training application with multimodal abilities. On the input side, the *touch* modality using NFC simplifies the interaction with the training device. On the output side, audio feedback improves the communication with the user while exercising when glancing at the screen is difficult. After the exercise, the visual modality is used to convey detailed information.

Activity Recognition

In the GymSkill use case, the user implicitly interacts with the device by performing the exercises. Unlike other mHealth applications that often confine to manual exercise logging (see the review in Section 2.2.2), activity recognition can reduce the amount of explicit interaction with the device. While prior research often investigated activity detection with wearable sensors [16, 94, 175, 185], we exploit the multitude of sensors that are already available in smartphones (see Section 2.1.2). We have exemplarily verified this approach for the use case of gym exercising, but activity recognition could also leverage other sports domains. As an example, evaluating acceleration and gyroscope data could classify the performed activity for endurance sport logging apps, making a manual setting of the performed activity in the app obsolete. The activity-related calorie expenditure can thereupon likewise be estimated. This results in a far more accurate determination of burnt calories after cardio training, as also pauses or speed changes are taken into account.

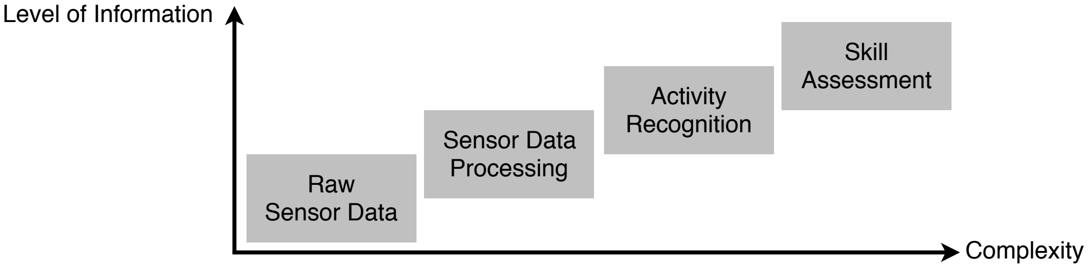


Figure 3.11: Consecutive levels of data acquisition and processing realized in the GymSkill application

Domain-Specific Benefit

Besides improving the usability of tasks that could theoretically be performed in other ways, MUSED interaction can also enable possibilities that would not be feasible otherwise. The skill assessment on top of the activity recognition (see Figure 3.11) leverages instruction quality and, as a consequence, fosters long-term motivation. As users can track their improvement with the individual sub-scores and the global score, they can see that regular training “pays off”. We believe this kind of feedback is more likely to provide intrinsic motivation that helps establishing continuous engagement, unlike gamification elements where prior studies have shown that people tend to lose interest after short periods of time [179]. Further, since individual aspects of the exercise performance can be assessed, instructions become tailored to the user. Hence, they are of more value to the user, since apps could identify aspects of exercises with potential for improvement or indicate body parts that need particular training. Based on this analysis, the app could suggest suitable exercises addressing “sticking points” and create a tailored training plan. Training assistance would become more efficient and help to reach goals faster.

Intelligent exercise assessment and monitoring is also relevant for elderly people, e.g., in an AAL context [93]. The smartphone application could be a reminder and a motivational factor to support physical activity, which is important for health risk reduction.

Chapter 4

University and Education

4.1 Problem Statement and Research Questions

As second large research area in mobile HCI, we look at a use case from the university and education domain. To be more specific, our main example in this chapter is MobiDics, a didactics tool which incorporates e-learning approaches to address teaching personnel in higher education. We first elicit the requirements and expectations of the target group for such a tool. Afterwards, we propose a solution to address the identified problems and goals with MUSED interaction. This research is motivated by prior work, which has already shown that the learning process with multimodal elements becomes more playful (increasing fun while learning) and more sustained [133, 182, 333].

In the second part of the chapter (see Section 4.4), we situate MobiDics in the context of three other systems developed in the context of our research. With the scenario we depict, we give an example for public-private interaction to illustrate how MUSED interaction blends with devices that are part of a university environment, as researched in the domain of intelligent environments [84, 280, 282]. Overall, the presented examples address people in the role of teachers (e.g., professors, lecturers, tutors, teaching assistants), as well as in the role of learners (e.g., students).

The systems presented in Chapters 3, 4 and 5 were not only chosen to show the benefit of MUSED interaction in a diverse set application areas. With our selection of examples, we also aim to span the design space in terms of application scopes. Unlike the personal training and ADL use cases in Chapter 3, the use case presented here is settled in a *semi-public* scope. This term shall denote that the degree of seclusion is not as high as at home, but also not as low as at an airport or a shopping mall (such environments will be subject of Chapter 5). Representing an intermediate state between private and public interaction, the following constellations on the public-private continuum are addressed in this chapter:

- Individual users interact over their (mobile) private devices with each other (mobile interaction; private interaction in public space – addressed in Section 4.3)
- Individual users interact over their private device with a public device (public-private interaction – addressed in Section 4.4)
- Many users interact with one public device or system (public interaction) one at a time or in parallel, e.g., with a public terminal, multimedia installation, etc. (addressed in Section 4.4)

The high-level research question answered in this chapter is:

- How can university employees and students benefit from MUSED interfaces in teaching and learning environments?

This chapter is partly based on papers we have published between 2011 and 2014 [168, 211, 219, 220, 234, 235].

4.2 Survey of Demand

As a starting point, we conducted a survey of demand for both the target group of educators and of learners. The goal of these requirements analyses was to assess which tools and possibilities the target group misses and which technical solutions they expect.

Requirements for Students and University Associates

We conducted an online survey among students and employees (academic and non-academic staff) at TUM (Technische Universität München) and LMU (Ludwig-Maximilians-Universität München). Participants were recruited via mailing lists (mainly for the staff), a university-internal discussion board, and by social networks. 93 subjects took part in the survey, whereof 60 were students. The survey investigated three parts:

- the current usage of university-related services,
- the usage of mobile Internet as enabler for mobile interaction, and
- the desired access to university services through *mobile* interaction.

In the survey, we presented a list of available online services at TUM. Not all of these services are dedicated for mobile device access (e.g., in that they provide a mobile website version or a native app). Out of the investigated services, the most frequently regularly used ones were the canteen menu (66%), the course management service (49%) and the room finder service (43%). When we aggregate regular and occasional use, room finder and course management are the post popular services with 84% and 81%. The results show that particularly the interactive tools are popular, while rather static websites such as the main web portal or the library site are known, but rarely used (24% and 25%). We subsequently asked subjects which services they would particularly like to access with mobile devices. Here, the highest interest is attested to a mobile room finder and an indoor navigation service (61% would "like to try", and another 15% would "maybe try" both systems). Furthermore (in the following, we use aggregated ratings of "like to try" and "maybe try"), survey participants would like to browse the library (57%), communicate with other students via Instant Messaging (50%) and locate them similar to Foursquare (29%), access the university website (46%) and be able to reserve rooms for learning (44%).

These ratings strongly differ from actual usage: For example, only 9% state to access the university website with their mobile device, and only 15% actually use the room finder service with their smartphone (although the service would probably be especially needed on the go). Only 6% actually send instant messages to fellow students, although half of all surveyed people would be interested in doing so.

How can this be explained? Given that the stationary usage of services is stronger than the mobile usage [168], we believe that there is a demand for mobile adaptations of university services. Current implementations are not well suited for small screens, touch-based interaction with a mobile device, and spontaneous use (some require a login, which is cumbersome with on-screen keyboards). Moreover, subjects show interest for novel applications that are not yet offered by the university, such as indoor navigation, locating students, and paying with the mobile phone at the university. We deduce a potential benefit for MUSED applications for the following reasons:

- They simplify and improve the interaction with university services on a mobile device, compared to "standard interfaces" that are not optimized for mobile usage.
- Sensors allow, e.g., context inference and thus better support services that are inherently context-dependent and predestined for mobile use, such as a room finder application. Sensor-driven interaction thus not only raises the mobile version of a service to the same level as the stationary interaction, but enables an even superior experience.
- MUSED interaction enables completely new applications where no stationary equivalent has existed before. This includes, e.g., on-campus mobile payment involving a NFC-capable phone, location-based search and navigation to on-campus points of interest (POI), enabled by an indoor localization system (see Chapter 5), or privacy-aware authentication to university services in a public space. Such services need to go along with appropriate user interfaces that keep their usage simple.

A comprehensive scenario complying to this vision is depicted in Section 4.4.

Requirements for University Educators

In a next step, in order to assess the requirements particularly for the MobiDics application we introduced at the beginning of this chapter, we interviewed people involved in university education (professors, lecturers, tutors, etc.) about their usage of didactic methods. We asked how they acquire knowledge about teaching methods and what problems they face in the practical application of teaching methods. The interview was conducted as online survey where 103 people (53 females, 50 males) took part. The average age was 33 years (SD = 9). The interviewees were mainly Ph.D. students (43%), professors (21%) and postdocs (15%), but also instructors and trainers, and they came from diverse subject areas (e.g., engineering, natural science, social science, or economics). 92% of subjects are smartphone owners and use it regularly, so that no technical barriers would prevent the use of a mobile didactics application.

From the free-text answers received in the survey, we distilled the following problems subjects have in the context of didactic method usage.

- They miss profound knowledge about the variety of didactic methods.
- Currently, people extract their information from didactic books, the Internet, and advanced training courses. For unexperienced docents, these information sources are experienced as too general; they do not address the very specific individual teaching situations. For example, when methods are described in the context of human disciplines, they are unsure whether they work for engineering subjects as well.

- Particularly young docents (e.g., student tutors, Ph.D. students) have little experience with teaching, and hence have had few opportunities to try or practice the use of didactic methods. Thus, they lack self-assuring feedback on the success of a particular method (by practicing it hands-on) for their individual teaching context.
- Since many docents are active researchers beside teaching, the remaining preparation time for courses is often sparse. If the benefit is unknown, the elaboration of new teaching concepts might be abandoned after a cost-benefit calculation, even before trying.

4.3 MobiDics – A Context-Sensitive Mobile Learning Tool

In the following, we describe the concept and implementation of MobiDics (short for Mobile Didactics), a context-based learning tool to address the problems unveiled in the requirements analysis. MobiDics has the following goals:

- Provide an overview of established didactic methods and their appropriate use in different teaching situations to activate students and to support them at every part of the learning process
- Achieve a deeper understanding of didactic methods by supplemental (and multimedial) content that is not available in traditional learning media like books
- Reduce the time for course preparation by more effective research of appropriate didactic methods in the planning phase
- Facilitate ad-hoc changes of the teaching concept in the classroom by adapting to the current context (such as a different equipment due to a room change, or the mood of the students during the lecture)
- Proactively suggest suitable teaching methods based on the individual teaching profile and personal preferences
- Enable individualized advanced training in self-study, focused on and adapted to the needs of the teaching person, independent of location and time
- Promote the exchange between docents on their experiences and on the success of having used a didactic method in a certain context
- Innovate professional education and training and offer mobile learning facilities as enhancement to traditional course-based training programs
- Reach target groups that would be addressable not as good without mobile learning

Pursuing these goals, MobiDics is a “train the trainer” application, with the ultimate objective to improve the quality of university teaching.

4.3.1 Didactic Background

For the creation of the learning content for MobiDics, we collaborated with PROFiL[69], Sprachraum[70] and the Centre for Learning and Teaching in Higher Education (Carl-von-Linde-Akademie/ProLehre[71]) which are professional training institutions at TUM and LMU.

MobiDics holds a collection of well-established, field-tested didactic methods, which represent a classic link between didactic background concepts and formulated educational goals in class. Well-considered use of specific didactic methods plays an important role in learning processes [188, 269]. Such methods can, e.g., activate students and thereby contribute to more profound and sustainable learning experiences [100]. At the university, where lessons and individual units are often longer and comprise more content than at schools, didactic methods have particularly high relevance in supporting the learning processes and their outcome. They can support individual learning phases (e.g., knowledge transfer, repetition, assurance of understanding), and thus increase the effectiveness of university education.

In MobiDics, didactic methods are organized based on a two-dimensional matrix where one dimension is goal-based, and the other is based on the social form.

Goal-Based Dimension

Learning goals at the university often have a cognitive character. In order to apply the acquired knowledge, often additional social and affective goals are required [100]. The first classification dimension thus supports multiple of these goals and is named after the German acronym *ARIVA* (also known as *AVIVA*). It was developed at TU Zurich [154] and classifies didactic methods according to the learning phase it which they can be applied. The *ARIVA/AVIVA* scheme comprises five phases:

- **Alignment** (German: "Ausrichtung"): Introduction and motivation of the learning content, raise of attention, match with the learner's world and experiences
- **Reactivation** (German: "Reaktivierung" or "Vorwissen aktivieren"): Activation of previous knowledge to provide a link for embedding the new learning content
- **Information** (German: "Information"): Active or passive knowledge acquisition, conveying of the learning content
- **Processing** (German: "Verarbeitung"): Deeper, more extensive and reflective processing of the content, e.g., by answering additional questions, integrating the learned content in larger contexts
- **Analysis** (German: "Auswertung"): Rehearsal of the learned content, answering of open questions that might have occurred in the processing phase, meta-analysis of the learning methodology

A "meta" category **Atmosphere** (German: "Lernatmosphäre fördern") complements the five phases. This category has the goal of making course members become acquainted with each other, creating team spirit and livening up the course. Each of the five *ARIVA/AVIVA* phases

[69] http://www.profil.uni-muenchen.de, Last accessed May 28, 2014
[70] http://www.sprachraum.uni-muenchen.de, Last accessed May 28, 2014
[71] http://www.mcts.tum.de/cvl-a/, Last visited May 28, 2014

is divided into sub-phases to allow an even finer adjustment of suitable methods to the desired learning outcome. The classification along this dimension ensures that each method incorporates and can be assigned to a clearly defined educational goal.

Social Form Dimension

The above phases are combined with different classroom formats, i.e., forms of cooperation and interaction of lecturer and students (the so-called social form). Examples of social forms are: work in pairs, small groups of three/four/five people, discussion in the plenum (entire class), or classic lecture style (also known as “chalk-and-talk teaching” [61] with no interactivity of the students). By respecting this dimension, teachers can, e.g., choose methods with alternating lecture, plenum, and group work phases to support maintenance of attention over longer periods of time.

Along these two structured dimensions, learning settings can be formalized and organized in form of didactic methods. This allows teachers a high flexibility to create learning situations appropriate for their needs. At the same time, they can be sure to create a sustained learning experience, as all methods are didactically well founded. When this dissertation was written, the MobiDics database contained approximately 100 didactic methods and is constantly growing.

4.3.2 Implementation

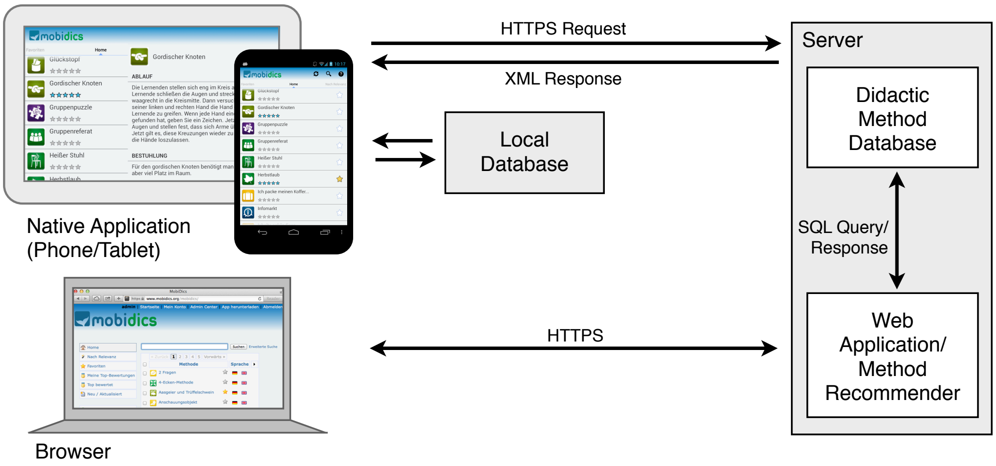
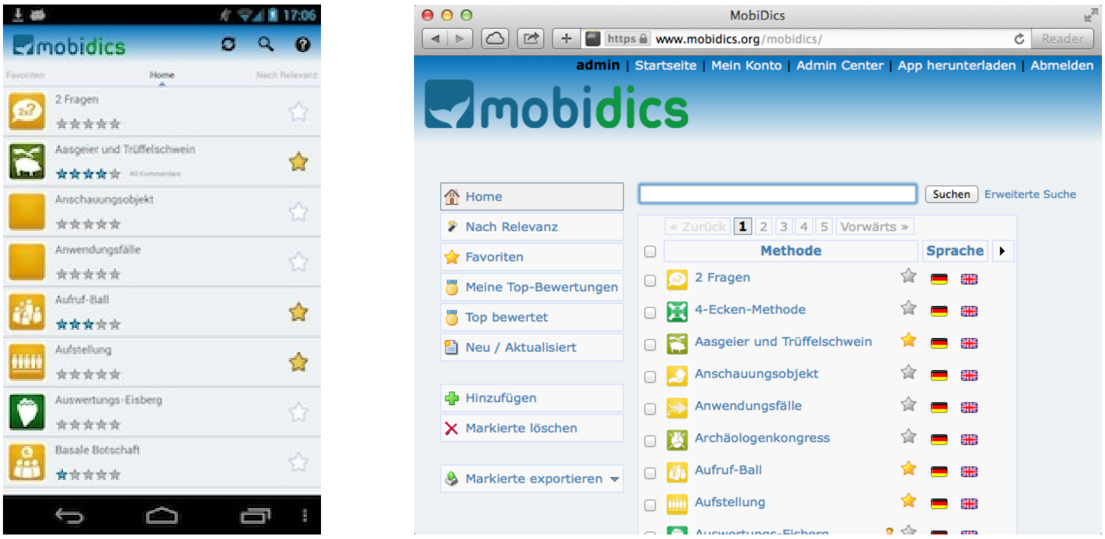


Figure 4.1: A schematic overview of the MobiDics infrastructure. MobiDics consists of a mobile Android application and a web interface, which both synchronize with the web server and database of didactic methods in the background.

System Structure

The MobiDics infrastructure consists of a server, a web interface and a mobile client application, which are illustrated in Figure 4.1. The didactic content is stored in a SQL database on the server and can be accessed via the native mobile application from a smartphone or tablet (implemented in Android), and via a web application, implemented with AJAX (Asynchronous JavaScript and XML) and PHP. Both native and web application provide access to the entire functionality of MobiDics, but differ in their user interface and interaction methods. While the browser-based interface focuses on classical mouse and keyboard interaction (also referred to as WIMP [326]: Windows, Icons, Menus, Pointers), the app interface makes use of interaction techniques like *swiping*, *pinching*, or *shaking* (a detailed description of the app's functionality follows in the next section). The web application, accessed from desktop or laptop computer, provides more screen space and thus allows a more comfortable navigation in non-mobile settings (e.g., in the office or at home). It is also the convenient way to enter longer portions of text, e.g., for commenting on methods or uploading own content. The mobile app supports a wide variety and a large heterogeneity of devices (smartphones, tablets of different sizes) and, thereby, also scenarios in which MobiDics can be used (e.g., in public transportation, during waiting times, or in class). The user interface layout adapts to different screen sizes and ratios for optimal use of the available space. A particular advantage of the mobile app is that it can be used offline. Each time a user starts the client application or logs into the web interface, she is authenticated with the server and changes are synchronized. Synchronization works in two directions: Both new methods and comments are downloaded to the client, and local changes are uploaded to the server and delivered to other users. When the client application is used offline, the most recent local state is used. Information is exchanged between server and mobile application in an XML (Extended Markup Language)-based data format over a secure HTTPS (Hypertext Transfer Protocol Secure) connection.

The screenshots in Figure 4.2 illustrate the user interface of app and web interface.



(a) Android application

(b) Web application

Figure 4.2: The Android and web version of the MobiDics system, showing the main menu with the collection of didactic methods.

Functionality

The heart of MobiDics is the collection of didactic methods, comprising extensive descriptions of their appropriate and correct execution, examples, visual support material, hints from didactic experts, and potential problems (e.g., what to do when students are not participating as intended). A gallery mode illustrates, e.g., constellations for group phases with sketches and optional interactive material (animations, videos). The didactic methods can be sorted by name, actuality, relevance (which is derived from the recency and the frequency of access), and popularity, based on ratings from other users. Besides the organization along the dimensions of social form and learning phase (see Section 4.3.1), each method is described and classified by further criteria (ideal group size, required material or equipment to apply the method, time the method consumes, applicability for different course types, etc.). Users can create their own methods and choose if they share them with the community or mark them as private. With SQL queries, even complex searches with multiple conditions can be performed quickly and efficiently in a large amount of data. Docents can use these filters to narrow down their method search to, e.g., *"a method for the reactivation phase that does not take more than 20 minutes, applicable for courses with more than 50 students"*.

Context-Sensitive Integration in Teaching and Learning Environments

As representative of sensor-driven interaction, MobiDics builds on context information teaching methods can be adapted to. Context in our particular case means the classroom (size, amount of freely usable space, available equipment like flip chart, whiteboard, etc.), course type, or time. The context plays an important role for the choice of didactic methods. For example, a method might require space for group activity or a certain equipment, or it might not be optimal for an evening lecture when students might be rather tired. With MobiDics, the docent can dynamically revise didactic concepts even during a short break in a lecture based on such context-specific conditions using the criteria-based classification. Furthermore, *context inference* can proactively respect such conceptual factors. The time and course type can be inferred from the personal schedule stored in the calendar or by an API to the university's room management platform (e.g., TUMOnline[72] at TUM). Such databases also contain information on room sizes and certain types of relevant equipment (e.g., whether chairs can be moved around for group work or the room has fixed rows). By an optional interface to this database, MobiDics can dynamically adapt its content to the available facilities for a planned lecture and context-sensitively react on, e.g., room changes. Moreover, MobiDics can retrieve a location estimate from the phone platform's location provider, so that it can be coupled with an indoor localization service (e.g., visual [225] or DECT [164] localization) or other location providers implemented on the smartphone.

Collaborative Learning and Exchange of Experience

MobiDics emphasizes collaborative learning, which is not yet naturally included in e-learning applications, although it has been proven effective in the real world [35]. In advanced training courses, participants often have very different backgrounds, previous knowledge, and teaching interests. Due to this heterogeneity, the course program hardly matches perfectly all participants' needs expectations. By contrast, the large user base of an e-learning system

[72] `https://campus.tum.de/tumonline`, accessed April 16, 2014

has a greater potential to find peers with similar interests or level of knowledge. MobiDics integrates possibilities for peer exchange in various ways:

- Users can upload own didactic methods and share their didactic knowledge with peers. This is an example where user-generated content complements the existing database and thus captures available knowledge of practical application of the theoretic content.
- Users can rate methods (with 1 to 5 stars), establishing a democratic, crowd-based "quality control" of user-generated content. The application of methods in different domains and subjects might yield different ratings, which makes them a benchmark for applicability in a certain discipline. If, e.g., the highest amount of good ratings for a method stems from engineers, this can tell with a certain reliability that this didactic method is suitable for engineering courses.
- Users can comment methods, which opens up discussions and professional exchange. Docents can share their experiences, ask questions or report solutions to problems or questions asked by other MobiDics users. Comments can be highlighted ("liked"), making useful contributions more visible for other users.
- MobiDics can by request recommend methods that might be useful to users based on their profile (taught courses, discipline), on their own and on other users' ratings of methods. The recommender algorithm is based on a combination of content-based [260] and collaborative filtering [183, 296]. The recommendations foster exploration of new teaching methods and at the same time keep up the likelihood that the newly tried concepts make didactically sense for the teaching situation.

Users of MobiDics have a profile that can (on a voluntary basis) contain information like the age, profession, discipline and taught courses, which is used for method filtering (*"methods applied in selected subjects or courses"*), or help estimating the expertise of a commenter through the user profile information (*"experienced professor teaching for 20 years"*).

Multilingualism

MobiDics seamlessly supports multiple languages (currently translated to German and English, and prepared for further translations), taking account for the fact that many universities (including TUM) offer courses in different languages in which, therefore, also the didactic methods should be available. Users can rank available languages to their preferences so that the app will display method descriptions in the language they master best. This is particularly important since didactic concepts and wordings are often difficult to translate and known under their original terms in the didactic community. MobiDics thereby fosters the communication and exchange of didactic concepts between docents of different mother tongues.

4.3.3 Evaluation

MobiDics was evaluated in several stages, pursuing different goals. We briefly summarize the evaluations and their respective objectives in the following.

Interest in Functionality

As a first step, we conducted an online evaluation, where an early version of the prototype was shown in a video. The goal of this survey was to get large-scale feedback on the general acceptance of a tool like MobiDics, and to learn which features are particularly important to the target group. 93 participants took part in the survey; they were the same participants of the requirements analysis described in Section 4.2. After having watched the video, 25% stated that they would use MobiDics "very likely", another 25% would use it "likely". 35% considered the usage rather unlikely, the rest would not use it (these statistics are based on subjects who stated in the survey to own a smartphone).

Furthermore, we asked subjects whether they would use particular features. Figure 4.3 shows the popularity of selected features of the prototype. Most popular were the method search based on various criteria (92%), the examples for each method (80%) and the multimedial explanations (69%). 63% were interested in the expert knowledge. These are particularly features unique to MobiDics, compared to alternative didactics information sources. These numbers are encouraging and we believe this shows the demand for tool support that goes beyond the possibilities of a paper-based method catalog.

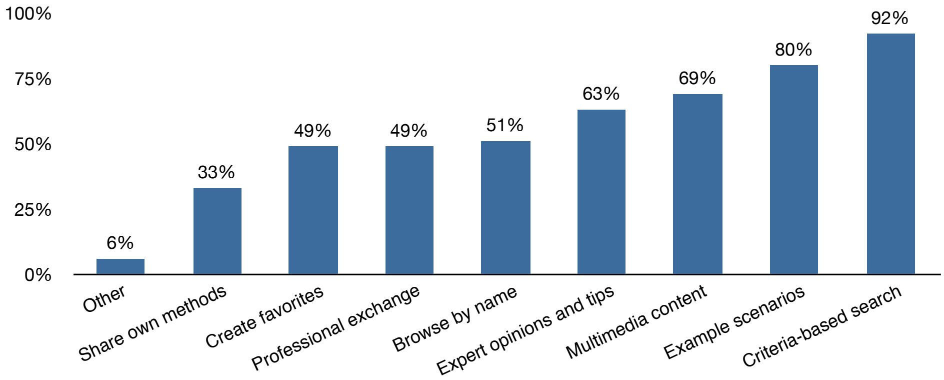


Figure 4.3: Selected features of MobiDics, ranked by popularity with potential users

Benefits for Teaching

In a second and a third step, we conducted two follow-up studies with a more sophisticated version of MobiDics. As both studies provide insights in the field of didactics, but do not focus on the (multimodal) interaction perspective, we only summarize them briefly in this work, and refer to the respective publications for more details [168, 250]. They are, however, important to mention, as they outline the success in achieving the (didactic) goals of MobiDics.

The first follow-up study intended to open up the design space for method systematization within MobiDics. In a qualitative survey, we determined factors that are, according to the participants, important for good teaching, and we asked participants about concerns in their own teaching. Based on the resulting discrepancy (subjects know about the benefit of didactic methods, but do not or rarely apply them), we generated ideas towards a "concern-based" approach of method selection. Example questions are: *"Which method can I choose when I am nervous?"* or *"I want exchange between students, but am concerned that students do not participate"*. This study was conducted with 47 docents in higher education, recruited from the academic training center at LMU, TUM, and other Bavarian universities. The results are described in detail in a journal paper we published in 2013 [168].

The other study was a long-term study to investigate the usage and the benefit of MobiDics in the real world and in everyday teaching. 22 subjects took part in this experiment. In order to monitor how MobiDics was used over the study period of four weeks, we used the self-developed SERENA logging framework [228], which we describe in detail in Section 7.3. Between start and end of the study period, a significant increase in knowledge about didactic methods and in self-confidence when applying them in class could be shown. This study was conducted in the course of a diploma thesis the author has supervised, which the reader is referred to for more details [250].

4.4 Integration in Learning and Teaching Environments

Let us now take a wider perspective and discuss the integration of the previously presented MobiDics application in a learning and teaching environment. While MobiDics addressed university educators' requirements, we now also refer back to the requirements for students that we determined in Section 4.2. We also come back to the distinction of interaction (public, private, and public-private), as we have made it at the beginning of this chapter. The vision we report on amalgamates these interaction types and also demonstrates the interplay of different interaction modalities.

Besides MobiDics, we incorporate three other systems in the scenario we describe, which are part of a technology-enhanced university environment.

- IRINA (Interactive Room INformation and Access system) is a touch-enabled digital door sign that is able to run various applications. Users can, for example, retrieve room occupancy information, reserve a room, or evaluate courses. IRINA is deployed in a pilot program[73] at selected lecture rooms at the Department of Electrical Engineering and Information Technology at TUM, and under active development in our research group and described by Roalter *et al.* [277, 278, 280].
- Ubiversity (Ubiquitous University) is a location-based social network that allows to locate friends within the university campus using a smartphone app. Unlike, e.g., Foursquare, the scope of Ubiversity is per definition limited to a local area and user group (university members), addressing data protection and privacy concerns. For further details, see our publication on Ubiversity [220].
- VMI Mensa[74] is a location-aware canteen menu application that displays the closest cafeteria or canteen, and informs on current menus and special offers, under consideration of individual requirements (accessibility options, ingredients, ...). The VMI Mensa

[73] `https://irina.ei.tum.de`, accessed June 6, 2014

[74] `https://play.google.com/store/apps/details?id=de.tum.ei.lmt.vmi.mensa`, accessed June 24, 2014

> application has been used as a vehicle to investigate large-scale app store deployment and update behavior of users [230] (see Section 7.5).

In order to illustrate the interplay of these systems, our scenario describes an exemplary university day, based on two *personae* [60], John and Emma. John is a 1st year bachelor electrical engineering student and unfamiliar with the university campus. He has already found some friends visiting the same lectures, with whom he often goes to lunch and works on exercise sheets. Emma is an assistant professor giving her first lecture in circuit theory in this semester. She has gained some theoretical knowledge about didactics and course planning, but she is curious how her concept will work out in practice.

> Monday morning, Emma has prepared an interactive part for her lecture, consisting of a group activity method where the groups shall summarize the results of short work assignments on flip charts. When she arrives at the lecture hall 20 minutes before the start of the course, she notices that the room was needed on short notice for an exam, and her lecture was moved to another room. She leaves a message at the digital door sign [278] of the lecture hall that the course has moved. In the replacement room, she finds that no flip charts are available, which would however be required for her planned group activity. She will thus have to re-plan a part of her lecture concept for today. She opens up MobiDics on her smartphone, and the application context-sensitively recognizes in which room Emma is currently in. Based on the room information database, MobiDics knows about the available equipment and suggests an alternative interactive method called “Buzz Group”, where students exchange their results in groups of two, without the need for flip charts.
>
> Meanwhile, the first students arrive at the lecture hall and see Emma’s note on the room change. Among them is John. As he is not yet familiar with the university building, he does not know where the new room is located. He touches the digital door sign to retrieve a route description to the new lecture hall. Next to the route description, a button is available to generate and display a QR code. John can scan the code with his smartphone. This brings the route description onto his device, so that he can easily find the new lecture hall. Using his visual navigation app [225], he finds easily his way to the new location.
>
> After the course is over, students have the possibility to give feedback on the digital door sign before they leave the lecture hall. John states that he liked the course and especially the interactive group phase, before he goes to lunch. John is a vegetarian, so he uses his location-based canteen app, VMI Mensa, to check for suitable menus around the campus. The app locates the cafeteria nearest to his current position on campus, and informs him that there a vegetarian pasta dish is offered.
>
> In the afternoon, John wants to find his friends to discuss the new Analysis exercise assignment together. He uses the Ubiversity app to locate his learning group. In the app, he sees that they have made a “check-in” at the computer lab to indicate their presence [166], so he goes there to find them. As the lab is quite noisy, the students leave the room to find a quiet place where they can discuss their homework. John remembers that the lecture hall was free after the course

> had ended. They go back to the lecture hall and check on the digital door sign that the room is free for the next two hours. Using the reservation button on the door sign, John books the room for him and his friends as a learning room. To confirm the booking process, John does not have to enter personal credentials on the door sign. A QR code is generated at the end of the reservation process that John scans with his smartphone. On his phone, he can securely authenticate with his university user account. The door is unlocked automatically and John and his friends can learn uninterruptedly.
>
> At the same time, Emma holds an afternoon C++ programming tutorial. As she notices that some of her students are sleepy and inattentive, she would like to reactive the class using a method to generate ideas for the upcoming project week. In a five-minute break, she uses MobiDics to spontaneously find a suitable activating method. By shaking her smartphone, MobiDics suggests her random methods, but all matching her specified didactic goal. The app automatically considers the requirements imposed by her course type, size, time (all deduced from her personal schedule), so that she can be sure the suggested methods are applicable in her current teaching setting.

This scenario shows us that the *modes of interaction* with the involved applications and services plays an important role for the practical applicability and integration in everyday university life. For example, the *Gesture* modality (shake gesture) for MobiDics can provide fast access to relevant content for docents when they are in a situation where time is spare. The *Scan* modality in the context of public-private device interaction (when booking a room with the IRINA door sign via a QR code) reduces privacy concerns when personal credentials are required [280]. We also note that context, in different forms (e.g., time and especially location), plays an important role. Since the location is determined by sensors (e.g., by WLAN localization in case of Ubiversity, or by GPS/cell localization in case of VMI Mensa), we can classify such systems as sensor-based user interaction. Here is a summary of the different types of sensor-driven and multimodal interaction that made possible the interplay of services in this scenario:

- Entering a message or giving feedback on the IRINA digital door sign using fingers on a touchscreen (*Touch* modality, public interaction)
- Getting location-dependent method recommendations from MobiDics (*sensor-based user interaction*, private interaction)
- Requesting directions to the new lecture hall from IRINA using the touchscreen, and transferring the route description to the smartphone by scanning a visual code (*Touch* and *Scan* modality, public-private interaction)
- Getting location-dependent lunch information with the VMI Mensa app (*sensor-based user interaction*, private interaction)
- Finding colleagues with the Ubiversity app (*sensor-based user interaction*, private interaction)
- Booking a learning room with the mobile phone via IRINA (*Touch* and *Scan* modality, public-private interaction)

- Requesting course-specific didactic methods from MobiDics, using touch gestures to navigate between methods and shaking the device to get random suggestions (*Gesture* modality, *sensor-based user interaction*, private interaction)

The determination of a user's position is here an especially important component, laying the basis for many services and applications using location information. We will therefore investigate the problem of indoor localization and navigation in more detail in the next chapter.

Chapter 5

Indoor Navigation

5.1 Problem Statement and Research Questions

This chapter focuses on MUSED interaction in the research area of indoor navigation. After pedestrian navigation outdoors has meanwhile become ubiquitous, localizing and guiding the user inside buildings is considered as one of current top tech trends[75]. Uses cases for indoor navigation applications can be, e.g., in airports, hospitals, hotels, shopping malls, convention centers, or museums. Such systems can not only provide users with directions to their departure gate, room, or store. Locating a device, associated with a user profile (and accordingly the user's goals and interests), is the basis for providing information about the environment, e.g., for informing on points of interest, museum exhibits, nearby offers, targeted advertising, and much more.

Reliable and usable indoor navigation is, however, not yet a scientifically solved problem. Out of the various indoor localization methods, we have identified *vision-based localization* as one particularly promising approach in Section 2.2.4. Using the visual modality to determine the location, this method works very similar to the way humans orient themselves and is a good example for multimodal and sensor-driven systems.

As the focus of this dissertation is not on the technical foundations for indoor localization, but on user interfaces and interaction methods, we describe suitable MUSED interfaces and interaction for visual localization. We will outline in this chapter that visual localization entails challenges that are not met by standard navigation system interfaces (we have given an overview in Section 2.2.4). We argue that multimodal elements in the user interface are especially well-suited to address these challenges and to adapt to the special properties of the visual localization technique.

The high-level research questions this chapter investigates are:

- How can the challenges of visual indoor localization be addressed by a MUSED user interface?
- How can multimodal elements be used to improve the user experience?

We report on our iterative research in the area of indoor navigation, incorporating four stages and several prototypes, and multiple user studies conducted online and in the real world.

[75] http://www.allaboutapps.at/2014/01/sieben-mobile-trends-2014-wearable-tech-indoor-navigation-und-mobile-banking-erobern-smartphone-co/, accessed June 10, 2014

This chapter is partly based on papers we have published between 2012 and 2014 [224–227, 229].

5.2 A Multimodal Interface Concept for Visual Indoor Navigation

The visual localization technique exposes some differences to other localization methods. Let us first identify what this means for user interfaces employed in conjunction with this technique. Subsequently, we introduce our novel UI concept, integrating augmented and virtual reality elements, and describe how they address the particularities of visual localization.

5.2.1 Challenges for User Interfaces

The UI challenges are closely related to the challenges emerging from the visual localization technique itself (see Section 2.2.4 for the explanation of the underlying principle). The success of visual localization depends on the quality of the query images. Ideally, they are crisp and portray a characteristic, unambiguous area of the environment, which we refer to as "distinctiveness" property. In that case, the query image can be matched well with the correct reference image, and the user's location can be precisely estimated. A query image is usually distinctive if it exhibits a high number of visual features. However, not all scenes the camera is pointing at during one's way through a building serve well for localization. We discuss two cases: either better query images can be found, or not.

The first case describes the situation where the environment offers suitable scenes for query images (e.g., characteristic signs, shop windows, posters, or exhibits), but the user does not target these areas with the mobile device. A reason for this can be that a typical pose of comfortably carrying a phone is about 45° downwards, which entails that the camera is rather facing the floor, but not corridors and rooms and the objects therein. Moreover, too much motion (due to walking, moving the device, or both) can add blur to the camera-visible scene. As a consequence, not enough visual features can be extracted for reliable visual localization. An ideal pose for visual localization would be upright as if taking a photo, as the "interesting" objects are typically found in eye height. While permanently maintaining this pose is inconvenient for the user, the UI could provide hints in which pose the device possibly "sees" better query images, if necessary.

In the second case, the user is at a location which actually does not exhibit unique visual features. This can, e.g., be the case in corridors with sparse texture resembling each other in different parts of the building. Query images taken in such areas are then indeed not sufficiently distinctive to detect the location. Temporarily, such route sections can be overcome with odometry and dead reckoning, but only with continuously decreasing accuracy. The associated localization and orientation errors (or a combination thereof) affect the navigational instructions in the user interface. In order to avoid misguidance, an improved method how instructions are presented should be found. As soon as a more distinctive region is in line of sight, it needs to be exploited to gain a new exact positioning. Active help of the user (directing the phone at such feature-rich regions when it is necessary to increase accuracy) is required. This active help could be demanded by the user interface.

In summary, the system should first try to ensure the best possible query images (i.e., add to more distinctiveness). If this is not possible, the system should cope as good as possible with decreased accuracy and still provide sufficiently working guidance instructions. The outlined problems show that without special adaptations from the side of the user interface, visual localization would in practice be likely to fail. The interaction concept can thus be seen as a central component and factor for the success of such systems. As we will show, MUSED interfaces can play a key role here.

5.2.2 Instruction Presentation

In the following, we present two main visualizations for instruction presentation on a handheld device (augmented and virtual reality), as well as additional sensor-based interface components, addressing the challenges outlined above.

Augmented Reality

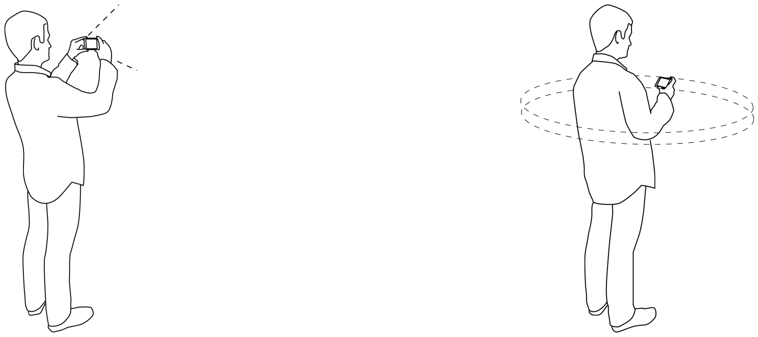
Augmented Reality (AR) enhances the video seen by the smartphone's camera by superimposing information. By this way of merging artificial elements with the real world, users can see navigation instructions, such as directional arrows, directly on their way. Users hold the phone as illustrated in Figure 5.1a and watch the overlaid video stream on the phone (video AR, unlike see-through AR like in HMDs), in order to see the augmentation directly on their way. Since this pose is desired for visual localization anyway (as previously discussed in Section 5.2.1), AR at first appears to be an obvious interface for a visual localization system. Further, AR visualizations have gained a certain level of familiarity through, e.g., commercial AR browsers[76]. However, it might be inconvenient to maintain the upright pose for long-term or frequent use (e.g., in unknown environments). This question will be investigated as part of the real-world user studies described in Section 5.3.3. Consequently, we propose an alternative visualization which does not rely on live video as it is the case for AR.

Panorama-Based Virtual Reality

This second visualization shows navigation instructions on top of a sequence of panorama images (comparable to Google Street View [6]), which we call Virtual Reality (VR) [313]. The panorama images used for the visualization are the previously taken reference images (they are available from the server without extra effort, as they are the basis for image matching in the course of visual localization). The individual images are merged to a panorama view on the mobile device.

Navigation arrows are directly rendered into the panorama, so that their orientation is fixed in relation to the virtual 360° view. We expect this to have several advantages (which we verify in the user studies presented in Section 5.3). First, no alignment of navigation instructions with live video (as in AR) is required. Thus, the device can be held in a more natural and comfortable way, as illustrated in Figure 5.1b). Second, we expect that fixated navigation arrows with relation to the panorama view are more reliable for navigation, as they even

[76] http://www.junaio.com, accessed June 12, 2014



(a) Augmented reality (AR) shows guidance as overlays on real-time video

(b) Virtual reality (VR) guides the user in a pre-recorded panorama view

Figure 5.1: Proposed augmented reality and virtual reality visualizations, depending on how the user carries the phone (upright or down) and on the location estimate's accuracy in order to improve the user experience and perceived quality of navigation instructions.

show the correct way in the panorama if the device's orientation estimate is not perfectly accurate. The panorama can manually be dragged around by the user for self-orientation, or aligned automatically according to the determined orientation. Furthermore, the frequency in which panoramas are updated can be adapted. In case no reliable localization estimate is possible, the update frequency can be lowered. Hence, we expect VR to be more robust than the more conventional AR view. The user interface is shown in Figure 5.17.

Multimodality

Visual indoor localization is inherently multimodal, as it uses the camera and orientation sensors for determining the location and presenting navigation instructions accordingly. However, also the user interaction with the our navigation application is multimodal. Besides the touching modality, we make use of the phone's compass sensor to automatically orient panoramas and navigation instructions. That is, guidance arrows will point in the correct direction even when the user turns around with the device. The panorama interface additionally allows manual manipulation. Users can drag the panorama view around with their finger. A "rubber band" effect can make the panorama snap back to the orientation provided by the compass when the user releases the view.

Furthermore, the phone's pose is used to choose the most appropriate visualization. When the phone is held upright (detected by the orientation sensor), the AR view is chosen. When the user holds the phone with the camera pointing downwards, the VR view is activated. The user can also manually switch between both modes.

5.2.3 Communicating and Ensuring Localization Accuracy

As motivated in Section 5.2.1, one goal of our UI must be to guarantee usable guidance, even if no good query images were captured by the user's device. The system should help improve localization accuracy, i.e., raise the lower bound for the quality of the location estimate. As outlined previously, we assume that a visual localization system can determine its location better when the device is held in an upright pose. Therefore, we show an indicator in case of low localization quality, prompting the user to actively point at regions containing more visual features. In other words, the purpose of the indicator is to make the user move up the phone along a vertical axis, bringing it from a pose as in Figure 5.1b to one as in Figure 5.1a. In case the user carries the phone in his hand (not currently looking at it) or in the pocket, the device vibrates to raise the user's attention (e.g., for feature indicators or for a turn instruction). The feedback modality (visual versus haptic) thus adapts to the user's and the phone's context.

Involving the user to help the system improve its position accuracy has already been used in other contexts for self-localization. For example, Kray *et al.* [174] asked users whether they can see certain landmarks from their point of view in order to perform semantic reasoning about their position.

We propose four different visualizations to make the user perform that "lifting" movement, which are depicted in Figure 5.2. All visualizations are based on the pose of the device, which we use as estimation for the suitability of the scene for successful localization.

- **Focus Change**: In analogy to a camera focusing on the motive, we use artificial focus change to guide the user towards a feature-rich area. Starting from a blurry scene, the image gets sharper the closer the user approaches a feature-rich area (see Figure 5.2a). This metaphor is inspired by an autofocus camera, motivating the user to find the "best" shot.

- **Textual Instruction**: A simple text hint is displayed, indicating to move the smartphone in a specific direction (see Figure 5.2b). A notification to raise up the phone appears until the pose is such that sufficient features are visible.

- **Color Scale**: A color-coded scale ranging from red (bottom/top, symbolizing few features) to green (center, symbolizing enough features) represents the number of distinctive visual features in the image (see Figure 5.2c). The color indicates the quality of the current scene for successful localization, so that the user should steer the indicator into the green area.

- **Bubble Level**: The metaphor of a bubble level is used to indicate the correct orientation of the phone. For an optimal position, the vial should be aligned in the center of the level (see Figure 5.2d). This can be achieved when the device is in upright pose, where a high number of features is most likely.

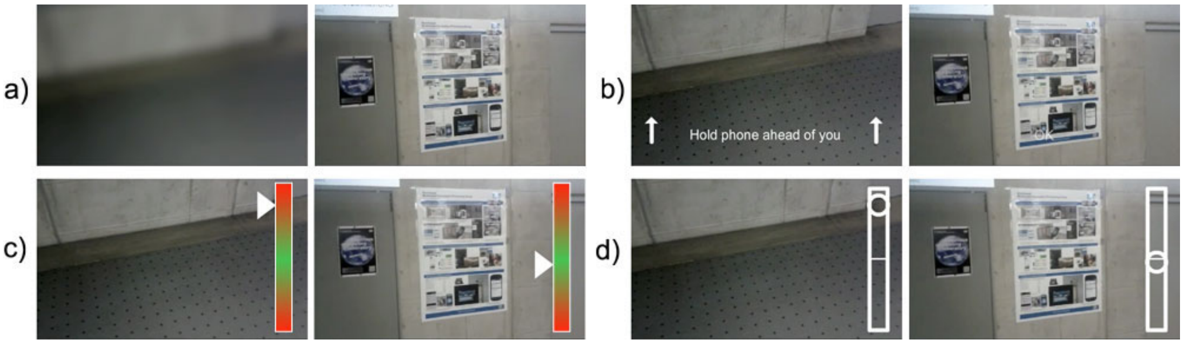


Figure 5.2: Proposed instructions to target the the phone at a feature-rich area: a) focus change, b) textual instruction, c) color scale, d) bubble level.

5.2.4 Highlighting Interesting Areas

Our concept also comprises a function to highlight areas/points of interest (AOI/POI) as part of the AR visualization. This fulfills two purposes.

First, certain objects in the environment can be marked as points of interaction, in the sense that they are entry points for context-based services. Thereby, the indoor navigation application is extended by the functionality of an augmented reality browser, such as Junaio[76]. One could imagine that users can see special offers by tapping a shop window, get background information on museum exhibits, or office hours by touching a door sign. Area highlighting can be realized technically with object recognition based on feature detectors, as we have demonstrated in Section 3.2, without the need for markers, tags, or other augmentations in the real world. Objects that are desired as points of interaction are often very distinctive, which makes them easy to identify.

Second, points of interaction often have a characteristic appearance, why they are well-suited as reference images for the localization algorithm. When a user focuses such a highlighted object with the camera, she helps at the same time improving the location estimate. Thus, interaction points have implicitly a similar effect as the four indicators presented previously. The user is not explicitly asked to change the pose of the device, but motivated to do so of one's own accord.

We conceived two visualization styles for highlighting objects. Our goal is to minimize the distraction of users when they do not want to interact with the interactive regions, but to still create enough awareness that these regions are noticed. One particular reason for distraction can be tracking problems, which cause the overlays to jitter [158, 332]. Our two proposed visualizations are depicted in Figure 5.3, where a poster is highlighted as an example. On the left, a frame visualization is used, while on the right a transparent overlay with soft borders is used. We hypothesize that inherent inaccuracies (and resulting jitter) can be hidden better when the visualization has no defined border.

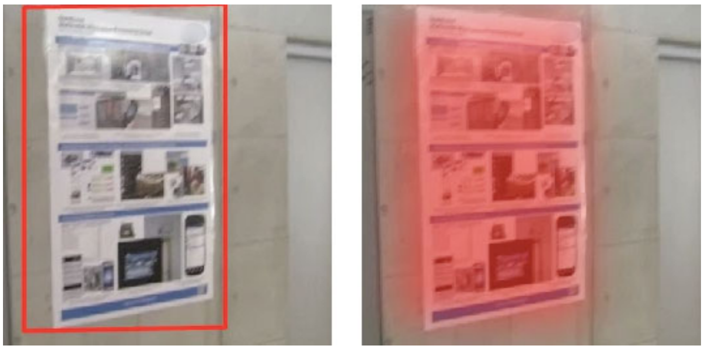


Figure 5.3: Highlighting of interaction points for context-based services. Instead of framing objects (left), we propose a soft border visualization (right) for less sensitivity to jitter and to reduce distraction of the user.

5.3 Comparative Evaluation of Augmented and Virtual Reality Interfaces

We evaluated the above described UI concept in two iterative studies. We began with an online study to gain initial feedback from a large number of users. The online study was based on videos with simulated navigation instructions, and images and animations illustrating the different interface versions and elements. Subsequently, we conducted a laboratory study, involving a prototype where we implemented the interfaces using the WOz approach [152]. The implementation of this prototype was informed by the results of the online study.

5.3.1 Research Questions

We formulate the following research questions, which are likewise used for the online and the laboratory study:

RQ1 Which concept (AR or VR) is preferable in terms of perceived accuracy?

We investigate whether and how the visualization influences the perceived accuracy, related to position and orientation, and the perceived quality of navigation instructions (note that this does not necessarily correlate with the actual technical localization accuracy). We hypothesize that VR, where the navigation arrow has a fixed direction in relation to the panorama, can improve the impression of accuracy. Especially when the system's location or orientation estimate is incorrect, we expect that the perceived reliability of the system is increased in VR, compared to AR.

RQ2 Which concept (AR or VR) is preferred by users?

We investigate the subjects' preferences for a particular visualization, which need to be taken into account as well for a convenient user experience. We hypothesize that AR might be less popular with users as a consequence of reduced perceived accuracy, and because of the fact that it the required upright pose is muscle-straining.

RQ3 Which visualizations could be appropriate to acquire sufficient visual features?

We investigate which of the visualizations presented in Section 5.2 has the strongest affordance character [247] for the user to raise the phone, in order to increase the likelihood of targeting objects with sufficient features for reliably localizing the device.

RQ4 Which object highlighting method is the best with regard to attention and distraction?

Here, we compare the highlighting visualizations *Frame* and *Soft Border* that we have introduced in Section 5.2.4.

5.3.2 Online Study: Accuracy Perception and User Preferences

Method

The study was conducted using an online questionnaire. The user interfaces and their operation in different conditions were presented to subjects in video demonstrations and images. The videos showed how it would look like if a user walked in a building using both the AR and VR system (see Figure 5.4). The videos allowed the subjects to compare the system's UI to what they would actually see (their environment).

To evaluate the effect of AR and VR visualizations on perceived accuracy, we used eight videos (four for each AR and VR) where artificial errors to the system's location estimate had been systematically added. The simulated route was always the same, but the navigation instructions shown in the video varied based on these errors. For example, the arrow pointed in the wrong direction (in AR), or a wrong panorama image was shown (in VR). We investigated four types of errors (conditions):

- **No Error**: All navigation instructions were correct; the pictures shown on the device matched the device's orientation and location.
- **Location Error**: An error was introduced as it would occur when the system's estimated location was incorrect. This type of error manifests in panorama images of an incorrect location (for VR), or incorrect turn instructions (for AR, e.g., when the system thinks of being next to an intersection where there is none). This error was induced twice in the *Location Error* condition.
- **Orientation Error**: An error was introduced as it would occur when the system's estimated orientation was wrong. This type of error manifests in incorrectly rotated panorama images (for VR), or incorrectly rotated arrows (for AR). This error was induced twice in the *Orientation Error* condition.
- **Combined Error**: Both location and orientation errors were introduced twice in this condition.

To each participant, eight videos were presented (within-subjects design). The order of conditions was counter-balanced using a Latin square design. Subjects were unaware of which error condition they were currently evaluating when watching the videos.

To evaluate the indicator visualizations (*Blur*, *Text*, *Color*, *Water Level*), we likewise used videos showing each visualization. The order was permuted using a Latin square design between participants.

Participants

81 subjects (39 females, 42 males), aged between 18 and 59 years (M = 28, SD = 9), took part in the study. They were recruited using the Mobileworks crowdsourcing platform[77]. Most subjects were infrequent navigation system users (43% use them only several times a month and 35% never). 18% use them often (several times a week) and 4 % very often (daily). Only

[77] https://www.mobileworks.com, accessed June 16, 2014

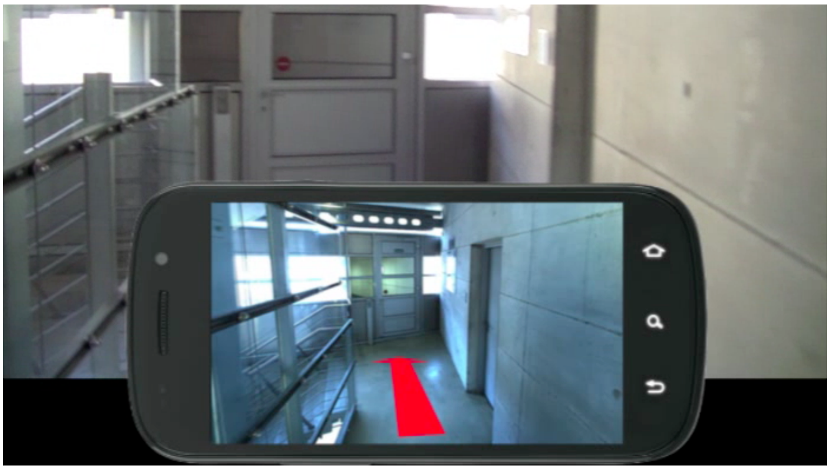


Figure 5.4: A screenshot from one of the videos used in the study, showing a mockup of the indoor navigation system. The upper part shows the simulated field of view, the smartphone screen the simulated navigation system output.

26% had used pedestrian navigation before, and 12% stated to have experience with indoor navigation. These indications suggest that users have no above-average knowledge on indoor navigation and can thus be considered as representative user basis.

Results and Discussion

All answers were given on 7-point Likert scales ranging from -3 (strongly disagree) to +3 (strongly agree). We used Friedman tests [104] and post-hoc Wilcoxon signed-rank tests [345] (with Bonferroni correction [90]) to examine effects of experimental conditions on perceived accuracy.

RQ1: Perceived Accuracy of AR and VR

Augmented Reality Figure 5.5 summarizes the evaluation of AR and VR with relation to the perceived accuracy. In the *No Error* condition, subjects felt that the system knew well their location (2.5, SD = 0.9) and orientation (2.4, SD = 1.0).

This perceived accuracy decreased in the error conditions. We found a significant effect both for the estimate how well the system knows the location (χ^2 = 77.10, df = 3, $p < 0.001$) and the orientation (χ^2 = 79.92, df = 3, $p < 0.001$). With an *Orientation Error*, subjects answered on average only with 0.8 (SD = 2.0) that the system was certain about their location, and with 0.2 (SD = 2.1) that it was sure about their orientation. For *Location Errors*, ratings were 1.7 (SD = 1.5) for the perceived location accuracy and 1.2 (SD = 1.8) for the perceived orientation accuracy. The perceived accuracy further decreased for the combined error condition. Here, the rating was 0.6 (SD = 2.0) for location accuracy and 0.4 (SD = 2.1) for orientation accuracy. Post-hoc tests showed significant effects between *No Error* and all error conditions. Likewise, *Combined Errors* were perceived worse than only a *Location Error*. However, there was no significant difference between the *Combined Error* and *Orientation Error* condition.

We also found a significant effect of conditions on the perception of correctness of the navigation instructions (χ^2 = 103.97, df = 3, p < 0.001). In the *No Error* condition, subjects averagely rated the correctness with 2.3 (SD = 1.0). With orientation and location errors, this rating decreased to -0.2 (SD = 2.0) and 0.4 (SD = 1.9), and with both error types together to -0.5 (SD = 1.9). Pairwise post-hoc tests revealed the significant differences between the *No Error* and all error conditions, but not between the *Orientation Error*, *Orientation Error*, and *Combined Error* condition.

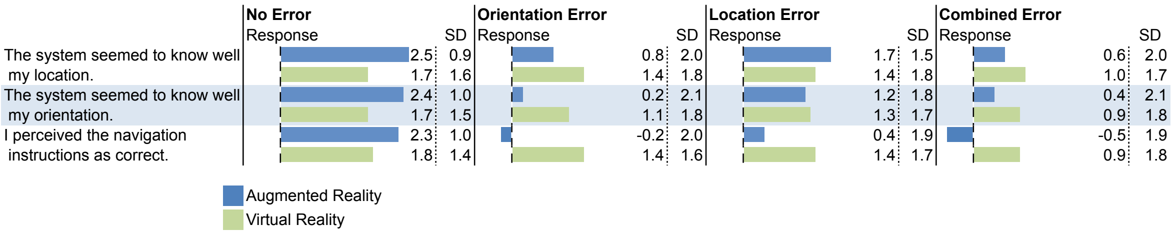


Figure 5.5: Perceived accuracy of virtual and augmented reality visualizations (agreement to statements on a 7-point Likert scale; -3 = strongly disagree, 3 = strongly agree). The diagram shows mean values and standard deviations (SD).

.

The gained insights from these results are twofold: First, instructions were perceived worse (i.e., less accurate) in the error conditions than with *No Errors*. Second, subjects had problems to distinguish the error types: In the *Orientation Error* condition, subjects perceived the orientation worse than the location (which was correct). In the *Location Error* condition, however, they had the same impression, although the location was actually more erroneous than the orientation here. This finding can actually be explained by the nature of orientation and location errors. An orientation estimation error clearly causes the navigation arrow overlay to point in a wrong direction. But also if the location is wrongly estimated, the arrow may likely point in the wrong direction (e.g., because the system interprets that the user has missed a turn). Thus, it is true that both types of errors may cause a similar effect of misoriented navigation arrows, so that the cause cannot be distinguished any more in an AR view. Such a situation is illustrated in Figure 5.6.

Virtual Reality In the *No Error* condition, subjects evaluated the perceived location and orientation estimate's accuracy with 1.7 (SD = 1.6 and 1.5). There were again significant effects on perceived accuracy (related to location: χ^2 = 19.36, df = 3, p < 0.001; related to orientation: χ^2 = 20.79, df = 3, p < 0.001). With the introduced *Orientation Error*, the rating slightly decreased to 1.4 (SD = 1.8) for the location estimate and to 1.1 (SD = 1.8) for the orientation estimate. In the *Location Error* condition, the perceived accuracy decreased to 1.4 (SD = 1.8) for the location estimate and 1.3 (SD = 1.7) for the orientation estimate. When both errors were combined, the perceived accuracy was rated with 1.0 (SD = 1.7) for location and with 0.9 (SD = 1.8) for orientation. Post-hoc tests revealed significant effects between *No Error* and *Combined Error* conditions (p < 0.01 for location and p < 0.009 for orientation), but not between the other conditions.

The perceived correctness of navigation instructions decreased from 1.8 (SD = 1.4) in the *No Error* condition to 1.4 in the single-error conditions (SD = 1.6 for orientation error and 1.7 for location error), and to 0.9 (SD = 1.8) in the *Combined Error* condition. This is a signifi-

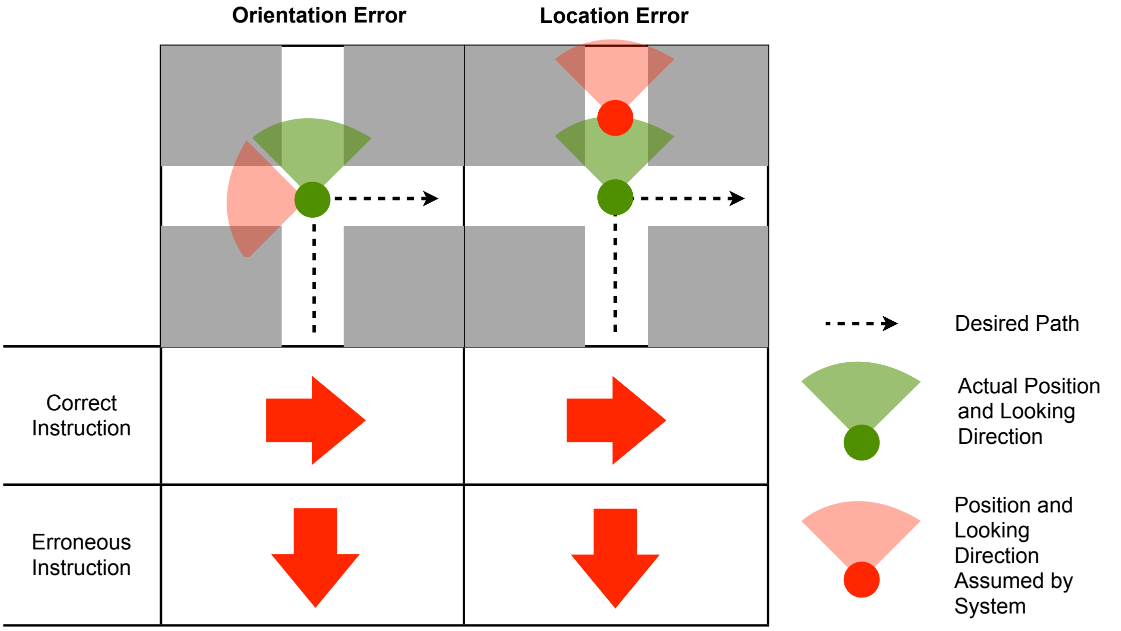

Orientation Error
Location Error
Correct Instruction
Erroneous Instruction
Desired Path
Actual Position and Looking Direction
Position and Looking Direction Assumed by System

Figure 5.6: Illustration of the effect of orientation and location errors. The user, symbolized by a dot, is standing in a junction (the cone symbolizes the looking direction). In the left case, the orientation is estimated by 90° wrong by the system; as a consequence, the navigation arrow erroneously says “go back” instead of “turn right”. In the right case, the location is misestimated so that the system assumes that the user has passed the junction. Thus, the navigation arrow, again, erroneously says “go back”. The two types of underlying errors cannot be distinguished by the user.

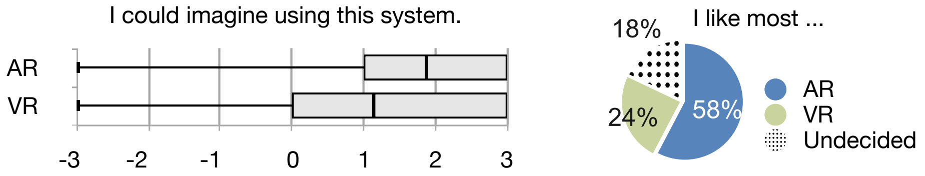
cant effect ($\chi^2 = 21.40$, df = 3, $p < 0.001$); however, a post-hoc test revealed a significant difference only between the *No Error* and the *Combined Error* condition.

Comparing the results of VR and AR, we can highlight the following two important findings:

- In VR, no significant differences of perceived accuracy between *No Error* and single error conditions could be observed. The AR system was rated significantly worse as soon as errors were introduced. This suggests that the VR visualization is more robust to errors than AR.
- Subjects rated the guidance in AR better than in VR in case of a perfectly working system, i.e., in the *No Error* condition ($p < 0.004$). However, in case of location or orientation errors, navigation instructions in VR were perceived significantly more correct than in AR ($p < 0.001$).

RQ2: User Preferences for AR or VR

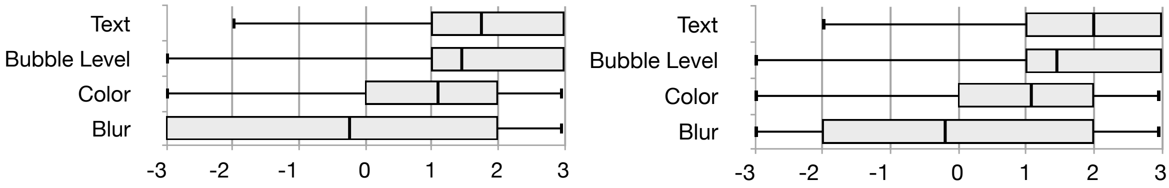
After having seen all videos, subjects could significantly better imagine using the AR system (1.9, SD = 1.3) than the VR system (1.1, SD = 1.8), $p < 0.05$. The high standard deviations indicate that the opinions were controversial, especially for VR. 58% liked AR most in the direct vote; VR was chosen by 24%, and 18% were undecided.



(a) Potential system usage

(b) Preference in direct vote

Figure 5.7: User preferences for the virtual and augmented reality system. Subjects preferred augmented reality (AR) over virtual reality (VR) navigation instructions. Answers given on a 7-point Likert scale; -3 = strongly disagree, 3 =strongly agree.



(a) Understandability of feature indicator visualizations

(b) Affordance to move device up to collect more features

Figure 5.8: User ratings of the feature indicator visualizations. *Text* and *Bubble Level* visualizations were evaluated significantly better than *Color* and *Blur*. Answers given on a 7-point Likert scale; -3 = strongly disagree, 3 = strongly agree.

RQ3: Feature-Rich Area Indicators

Subjects evaluated the understandability and the affordance character of the four indicators (*Text*, *Water Level*, *Color* and *Blur*) presented in Section 5.2.4. We found that the indicators had a significant effect on understandability (χ^2 = 38.99, df = 3, p < 0.001) and on affordance (χ^2 = 58.91, df = 3, p < 0.001). The results are visualized in Figure 5.8.

Subjects found *Text* and the *Bubble Level* level most understandable (no significant difference between these two). Subjects responded that the meaning was clear on average with 1.7 (SD = 1.5) for *Text*, and that of *Bubble Level* with 1.5 (SD = 1.6). *Color* was evaluated with 1.1 (SD = 1.7). The understandability of *Blur* was below average and showed a high standard deviation (-0.2, SD = 2.2). Post-hoc tests showed significant differences between *Text* and *Color*, and between *Blur* and the other visualizations (p < 0.05).

Similarly, *Text* (2.0, SD = 1.4) and *Bubble Level* (1.5, SD = 1.5) were evaluated best regarding their motivational effect to raise up the phone (affordance character). *Color* (1.1, SD = 1.7) and *Blur* (-0.2, SD = 2.1) were, again, rated significantly worse in this discipline (p < 0.05).

RQ4: Object of Interest Indicators

We evaluated the two visualizations for highlighting objects of interest as presented in Section 5.2.4. The method shown left in Figure 5.3 will be called *Frame* in the following, while the method depicted in the right will be called *Soft Border*. The results are visualized in Figure 5.9.

There was no significant difference between how *Frame* and *Soft Border* draw the subjects' attention on the highlighted object (p > 0.1). *Frame* was evaluated significantly more conve-

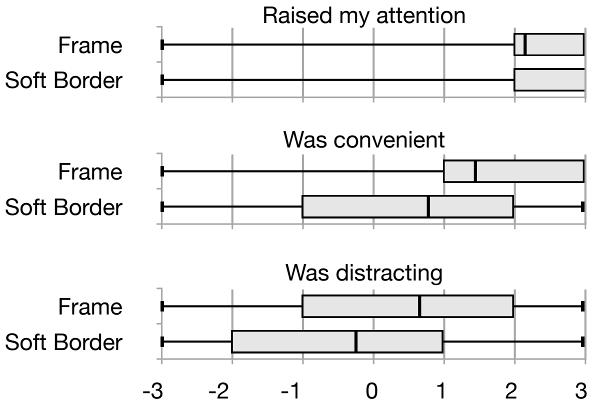


Figure 5.9: User ratings of different highlighting and tracking visualizations of interesting objects. At similar level of attention, a soft border highlight (cf. Figure 5.3) was perceived as less distracting than a border around the object, but subjects also found it less convenient. Answers given on a 7-point Likert scale; -3 = strongly disagree, 3 = strongly agree.

nient (1.5, SD = 1.6) than *Soft Border* (0.8, SD = 1.9), $p < 0.05$. On the other hand, subjects attributed a significantly lower distracting effect to *Soft Border* (-0.2, SD = 2.0) that to *Frame* (0.7, SD = 1.9), $p < 0.05$.

We showed subjects an additional video where the background video was desaturated (black and white), and only the highlight was in color. We hypothesized that this contrast could further focus attention to the object and thus be beneficial. Results showed however no significant differences in attention, convenience or a distracting effect between colored and desaturated backgrounds.

Discussion, Lessons Learned, and Limitations

Accuracy Perception and User Preference for AR/VR

This first study has already shown that AR and VR show their strengths in different domains. VR is less impacted by localization inaccuracies, why it was perceived as more reliable for incorrect location and orientation estimates. When the VR panoramas are slightly translated or rotated, people can still match them with the environment. What is important is that the navigation arrow is still always correct *in relation to the VR view*, even if it is not in relation to the real world.

While AR overlays get wrong much faster, even for slight inaccuracies, this interface is more natural in case there are no location or orientation errors. This intuitiveness probably leads to the fact that users favored AR so strongly in the direct vote. This preference is remarkable, as one would have expected that subjects prefer the visualization which gives the more accurate impression. We have two possible explanations for this result. First, AR probably appeared in the mockup as the more elegant solution, compared to a “flip book” impression of VR. Given the fact that real-world localization (unlike in the mockup) will never be completely exact, this preference might change in a hands-on study. Second, we hypothesize that *in situ*, users will take into account when determining their preference that they need to carry the phone in a more uncomfortable pose for AR to work. Such physical usage factors cannot be determined in an online study, so that this effect will have to be investigated in a hands-on study as well.

Understandability of Indicators and Object Highlighting

Our analysis of *feature indicators* addressed the important point of creating awareness for how well a scene serves for localization and how the user can assist the system to improve accuracy. Sufficient salient features in the image are crucial for reliable vision-based localization. The *Text* and *Bubble Level* metaphors were rated as most understandable and motivating to raise up the phone to feature-rich areas, so that we will focus on these visualizations for a hands-on study. In a next step, it will have to be verified in a real-world study whether such visualizations are actually an incentive to focus on feature-rich areas in eye height.

For highlighting objects, the *Soft Border* method reduces distraction and thus might interfere less with the navigation task. Likewise, the *actual* effect under real-world conditions will have to be investigated in a hands-on study.

Panorama Update Frequency

For some visualizations, we evaluated minor variants that we presented to our subjects in additional videos for evaluation. We provided different versions of the VR system where we modified the frequency in which panoramas were updated. The system used for comparison against the AR system (which we discussed previously) updated the panorama about every second. In addition, we provided a version with faster update rate (every 0.5 seconds) and with slower update rate (every 2.0 seconds). The one-second frequency was appreciated slightly more (1.0, SD = 1.7) than faster (0.8, SD = 1.8) or slower panorama changes (0.6, SD = 1.7). Only the difference between medium and slow transitions was significant (Student's t-test [318] with $p < 0.05$). Keeping this in mind, we will keep update rates rather slow for the implemented version.

5.3.3 Experimental Study: Real-World Validation

With a follow-up experimental evaluation, we aim at answering the following questions that remained open in the online study.

- Subjects rated the VR visualization to be more reliable, but still preferred AR in the online study in a direct ranking. In the experimental study, we will investigate whether hands-on tests yield different results regarding perceived reliability and user preference. There are several differences to the online study: First, the interfaces are evaluated in an interactive mode rather than by passive videos. Using the interface thereby becomes a secondary task (while walking in a building). Second, subjects have to carry the phone in their hands, so that they experience the comfort of the AR and VR poses. Third, the UI is experienced on the small screen of a mobile device, instead of a full-size computer monitor.
- While in the online study, AR and VR only could be evaluated separately, in the experimental study, we can combine both modes in a prototype to see which mode subjects actually use more frequently in a navigation task.

- In the online study, we evaluated the understandability of additional UI elements (indicators to raise the phone up), but not their actual effectiveness. Only an experimental study can tell if these elements really lead to more detected features and thus to improved localization. Similarly, the *Frame* and *Soft Border* highlighting visualizations were only evaluated based on mockup videos, but not with real object recognition.

Implementation

Client Application

For the experimental study, we implemented a prototype application based on the Android 2.3 platform[78]. The app incorporates the AR and VR visualizations and additional UI elements presented in Section 5.2; the realization in the prototype is shown in Figure 5.17 (left image). A button on the top right allows to switch between VR and AR with a button on the top right of the screen. The system can also switch modes automatically based on the gravity sensor readings. In an upright pose as in Figure 5.1a, the system switches to AR; in a pose as in Figure 5.1a, the VR visualization is selected. The threshold angles were set to empirically determined values of a 35° inclination for switching to AR, and a 30° inclination for switching back to VR.

The navigation interface was implemented with OpenGL ES 2.0. For VR, it displays 360° panorama images of key locations, surrounding the user's point of view. The navigation arrow is drawn on top of the panorama view at the correct location. For AR, the directional arrow is anchored to virtual "key point" locations similar to VR, except that it is overlaid on live video from the rear camera. For both AR and VR, the magnetic field sensor is used to auto-rotate the visualization, accounting for the measured device orientation. In VR, panoramas can be manually rotated by drag-and-hold; lifting up the finger re-enables auto-rotation.

As feature indicators to motivate users to raise the phone up, we chose a combination of the *Bubble Level* metaphor and a *Text* hint, as these two were evaluated best in the online study. The indicator can either pop up automatically when the number of visible features falls below a definable threshold. For the automatic trigger, we detected the number of FAST [286] features in the camera's live image, using the OpenCV framework for Android[79]. The anticipated position of the device (90° angle) is determined by the phone's gravity sensor.

The object highlighting function was realized using an image processing pipeline as depicted in Figure 5.10. We implemented the *Frame* and *Soft Border* visualization using two slightly different methods. For *Frame*, a contour detection is applied after edges have been enhanced by a Canny edge detector [43]. The most significant contour is selected and enhanced with a bounding rectangle. For *Soft Border*, we apply a FAST feature detector [286] and count features in local subareas of the image. The area with the most features is highlighted with a semi-transparent rectangle.

[78] At implementation time (July 2013) still over 33% of devices ran Android 2.3 or lower, see `http://developer.android.com/about/dashboards/index.html`, accessed September 23, 2013

[79] `http://opencv.org/platforms/android.html`, accessed August 13, 2014

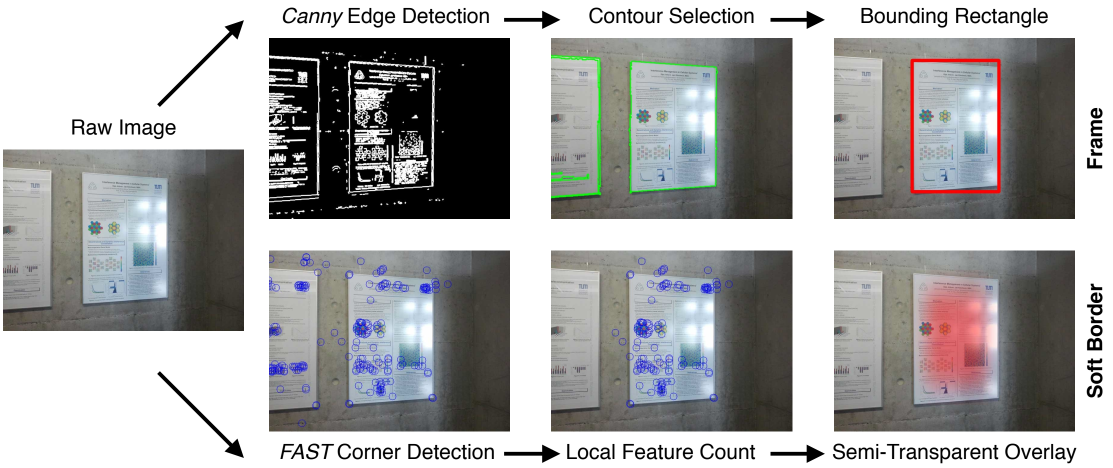


Figure 5.10: General proceeding for detecting and highlighting objects with two different visualizations: a soft border overlay, supposed to be less distracting (left), and a rectangular frame (right).

Wizard-of-Oz Testing

The view components of the prototype are loosely coupled with the control logic, as defined by the Model-View-Controller (MVC) [39] pattern. This would, e.g., allow to couple the UI with a live underlying localization system, which basically would perform image matching with a database on a server, and return the location of the most similar reference image. However, for our study we implemented the navigation mechanism with a WOz approach [152], motivated by the advantages of WOz testing as outlined in Section 7.2. This proceeding especially allow us to modify the accuracy of position and orientation estimates, in order to reproduce the different conditions of the online study in the experiment. Our evaluation setup consists of two Android apps: the just presented client app (running on the test subjects' phone), and a WOz app for the experimenter (see Figure 5.11a).

With this app, the experimenter sends location information to the subject's device at the desired position of the route. The client displays these data as location arrow overlay (in AR) or panorama image with navigation arrow (in VR). The WOz app holds the data for several predefined routes used in the experiment, which contains the panorama photos and the associated arrow directions for each key point. The experimenter just has to hit a button to send the next correct instruction to the subject, but can also deliberately trigger localization and orientation errors.

Hypotheses

For the experimental evaluation, we refer back to the research questions RQ1–RQ4 formulated in Section 5.3.1. Informed by the results from the online study, we now formulate the following hypotheses H1–H4 based on these research questions. We extend RQ1 (addressing the perceived accuracy of VR and AR) by the aspect of efficiency of the two visualizations (which was not measurable in the online study). As we can now measure the navigation time, we insert a hypothesis H1(b) as follows.

H1(a) Subjects perceive VR to be more accurate in case of localization errors than the AR interface.

H1(b) VR is more efficient than AR in terms of navigation time to the destination.

H2 Subjects prefer the VR interface over the AR interface.

H3 The *Bubble Level & Text* visualization leads to in average more visual features per frame.

H4 The *Soft Border* highlighting method is less distracting than the *Frame* highlighting method.

Method

We conducted three hands-on experiments to investigate the formulated hypotheses. In all experiments, subjects used a Samsung Galaxy S II (4.3-inch screen, 8 megapixel camera). The WOz app operated by the experimenter ran on a Samsung Nexus S (4-inch screen). Both devices had a screen resolution of 480×800 pixels and were running Android 2.3.

Experiment 1: Comparison of VR and AR (H1 and H2)

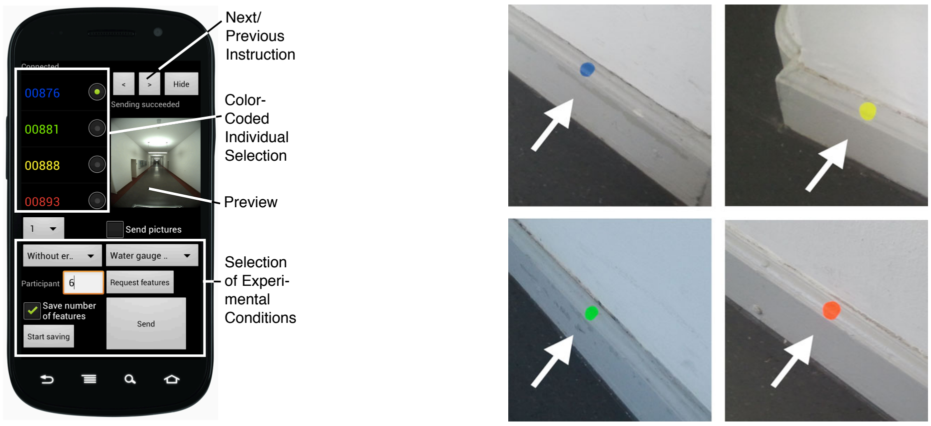
Subjects performed a navigation task inside the main building of TUM on a path of 220 meters length (see Figure 5.12). The route was chosen to show sufficient complexity. The accuracy of the system's location estimate was varied in four conditions by the experimenter (*No Error*, *Position Error*, *Orientation Error*, *Combined Error*), both in AR and VR mode. Consequently, each user traversed the path eight times. The path was the same in all conditions for better comparability, but the order of conditions was counterbalanced with a 4×4 Latin square to weigh out learning effects. Subjects were asked to rely only on the given instructions, so that they could not be sure whether the path would not vary.

Navigation instructions were fed into the subject's phone by the experimenter who walked about one meter behind the subject, according to the WOz approach. Colored labels in the app and on the skirting board (see Figure 5.11b) helped the experimenter to trigger the correct image at the same locations for each participant during each experiment.

In error conditions, correct panoramas and arrows were replaced twice by short sequences of misplaced (*Position Error*) and misoriented instructions (*Orientation Error*). The locations where the errors were introduced were the same for all participants. Start and end time of each run (from receiving the first panorama until reaching the destination) were measured by the prototype. Users were asked to "think aloud" [328] while using the system and answered a questionnaire after each run.

Experiment 2: Effect of Feature Indicator (H3)

In order to evaluate the feature indicator visualization (using the *Bubble Level* metaphor combined with text), we constructed a setup as it would occur in a self-contained navigation system. It is likely that, from time to time, a relocalization procedure would be required, in which the user is explicitly asked by the system to direct the phone to a feature-rich area.



(a) WOz app

(b) Markers

Figure 5.11: The WOz app for controlling visualizations on the subject's device and simulating localization errors (left). Markers in the corridors (right) helped the experimenter to trigger visualizations at identical locations for similar experimental conditions.

Subjects performed another navigation task, similar to the previous experiment. Three times during the walk, the experimenter triggered the *Bubble Level* visualization to appear on the subjects' device. As soon as subjects raised the phone until the bubble was centered on the scale, the indicator disappeared and a location update (i.e., the correct arrow/panorama) was displayed.

In the experiment, the prototype switched automatically between the AR and VR visualization based on the phone's inclination, as described in the Section 5.2.2. Subjects could freely decide how to carry the phone. Therefore, the experiment was also a test which visualization (and inherently, which pose) was chosen more frequently by subjects.

We logged the inclination of the phone (whether it was carried down or upright), whether the feature indicator was currently shown or not, as well as the number of detected FAST features (all in one-second intervals). After the experiment, users answered a questionnaire.

Experiment 3: Object Highlighting Methods (H4)

In this experiment, subjects tried out the *Frame* and the *Soft Border* visualization (see Section 5.2.4). As example object of interest, we used a poster at which subjects pointed, in order to evaluate the two highlighting methods. We had tested beforehand that the poster could be robustly recognized with both algorithms (see Figure 5.10). Afterwards, subjects answered a questionnaire.

Participants

12 people (11 males, 1 female) between 23 and 27 years (M = 24, SD = 1.3) participated in the study. Most subjects were students, thus matching the potential target user group for on-campus indoor navigation. None of the participants were involved in our research project. No compensation was paid. The experimental design of all three experiments was within-subjects.

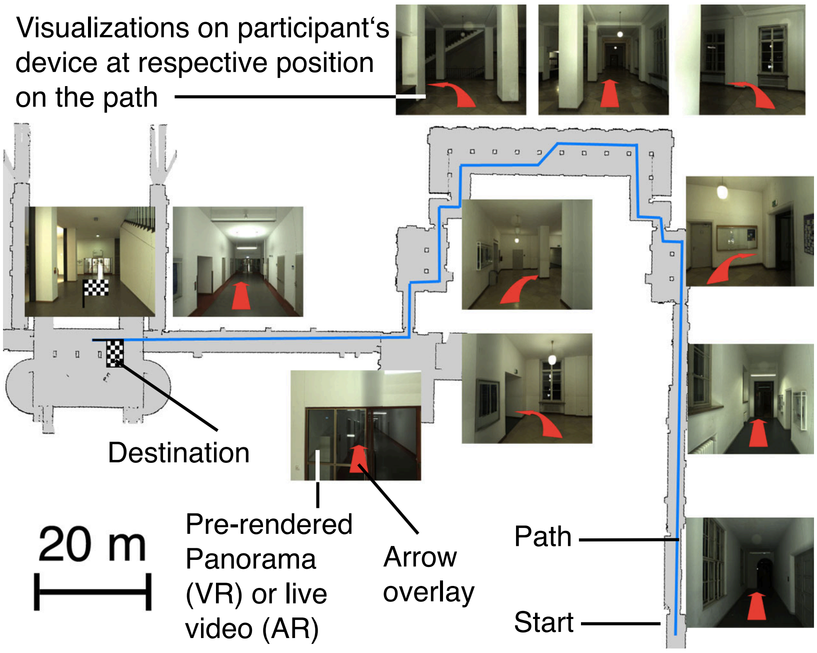

Visualizations on participant's device at respective position on the path
Destination
20 m
Pre-rendered Panorama (VR) or live video (AR)
Arrow overlay
Path
Start

Figure 5.12: The indoor path used for the navigation task in the study (220 meters), alongside with some sample images and route instructions as they were displayed on the subjects' phone.

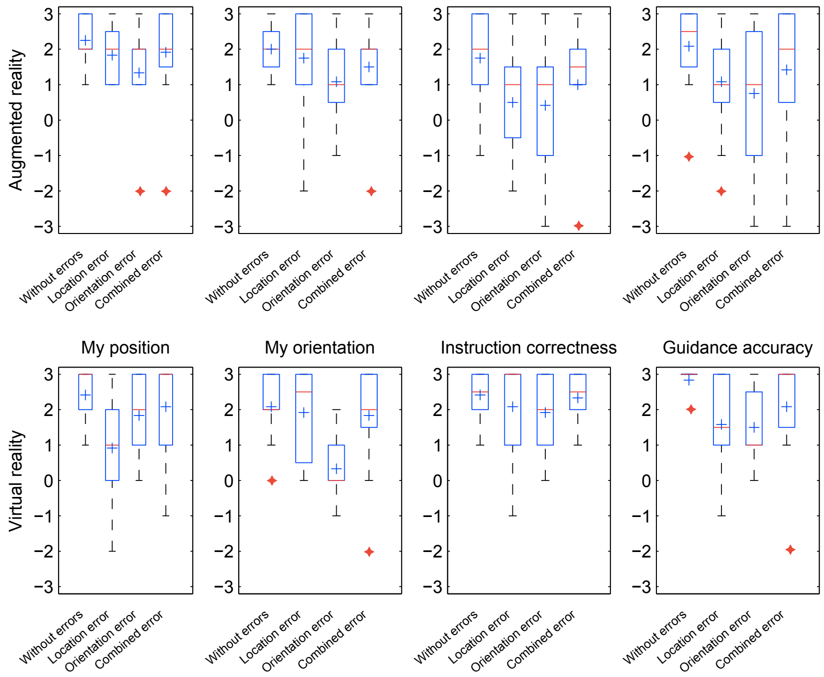
Results and Discussion

H1(a): Accuracy Perception

Subject rated the perceived accuracy in the conditions *Without Error*, *Position Error*, *Orientation Error* and *Combined Error*. Subjects were presented the following statements: *"The system seemed to know well where I am"* (relating to the position estimate), *"The system seemed to know well in which direction I am looking"* (relating to the orientation estimate), *"The navigation instructions were always correct"* (relating to the perceived correctness of individual instructions), and *"Overall, I found the guidance accurate"* (relating to the general guidance accuracy). Agreements to each statement were indicated on a symmetric 7-point Likert scale where -3 corresponds to "strongly disagree" and +3 to "strongly agree". Figure 5.13 summarizes the responses in box plots. In the following, we use medians and Wilcoxon signed-rank tests to report the results, using a significance level of $\alpha < 0.05$.

Let us first see if subjects were able to identify the introduced position and orientation errors. Both in VR and AR mode, subjects perceived a difference in the guidance accuracy between the *No Error* and the error conditions. Both for position accuracy (AR: W = 15, p = 0.037; VR: W = 28, p = 0.021) and orientation accuracy (AR: W = 19.5, p = 0.073; VR: W = 55, p = 0.005), Wilcoxon signed-rank tests yielded significant differences compared to the *No Error* condition (except for orientation accuracy in AR).

However, only in AR, we observed that these errors had a significant effect on perceived correctness (p = 0.015 in case of position errors and p = 0.034 in case of orientation errors, each in relation to the *No Error* condition). With VR, the differences in perceived correctness were not significant. Consequently, the rating of perceived correctness of instructions was



Mode	Condition	Time m:ss	Avg./ Mode
AR	No Errors	2:52	3:04
	Position	2:53	
	Orientation	3:23	
	Combined	3:07	
VR	No Errors	2:36	2:39
	Position	2:41	
	Orientation	2:44	
	Combined	2:36	

▭ Interquartile range (25%-75%)

+ Mean

— Median

⊢ – – Min/Max

✦ Outlier

My position: The system seemed to know well where I am.
My orientation: The system seemed to know well in which direction I am looking.
Instruction correctness: The navigation instructions were always correct.
Guidance accuracy: Overall, I found the guidance accurate.

Figure 5.13: *Left:* Perceived guidance accuracies in experimental conditions of AR and VR interfaces. The box plots visualize the level of agreement to the statements on the bottom right. (on 7-point Likert scales ranging from -3 to +3). *Top right:* Task completion time using VR and AR. In AR, subjects on average took 25 seconds longer, and differences between conditions were higher.

significantly higher for VR than for AR: With *Position Error*, rating medians were 3 for VR and 1 for AR (W = 6, p = 0.030). With *Both Errors*, medians were 2.5 for VR and 1.5 for AR (W = 3.5, p = 0.023). Only with *Orientation Error*, no significant differences could be observed (VR: MD = 2; AR: MD = 1; W = 4.5, p = 0.065). Those results indicate that VR is generally considered to be more accurate than AR (which supports **H1(a)**).

H1(b): Efficiency

Subjects reached the destination in average 25 seconds faster with VR (averagely 2:39 min:s for the 220 m path) than with AR (averagely 3:04 min:s). This is a significant difference according to a paired sample t-test (p = 0.002) that confirms **H1(b)**. The different error conditions did not have a significant effect on navigation times in VR. However, with AR, differences between conditions were partly significant: Subjects were here slower in the *Orientation* and *Combined Error* condition than in the *No Error* or *Position Error* condition (see top right table in Figure 5.13). This signifies that AR can, in case of (particularly orientation) errors, be inferior to VR in terms of efficiency.

H2: Convenience and User Preference

Asked for the preferred system, 50% decided for VR, 33% for AR, and 17% were undecided (supporting **H2**). This strong tendency is presumably not only grounded in the quality of navigation instructions, which was perceived to be better in VR, but also in the convenience

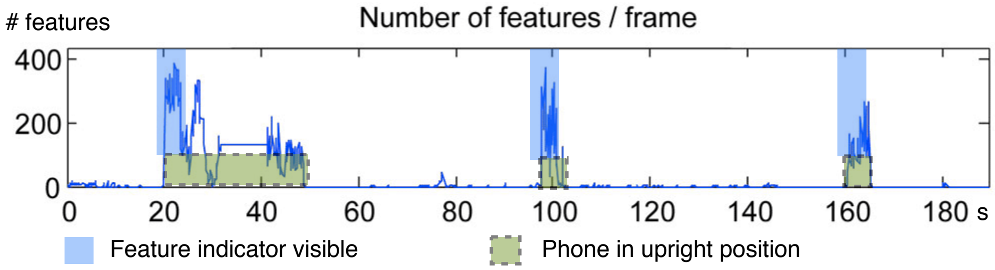


Figure 5.14: When the feature indicator is visible (light blue), users move the phone up (green) and more visual features are detected per frame. This diagram exemplarily shows one subject's data.

when using the system. The answers whether subjects find carrying the phone convenient were significantly better in VR (MD = 2) than in AR (MD = 0), W = 0, p = 0.009. The required upright position for carrying the phone in AR was physically constraining. One participant said that it could work *"well for 200 meters, but not more"*. Most subjects also had a feeling of embarrassment to pass by other people in that pose, because others might fear being recorded. This problem was not given in VR, because the camera in that case points towards the floor.

H3: Feature Indicator

While the indicator was visible, the number of detected features per frame rose from averagely 42 to averagely 101. In empirical trials, we found that reliable localization works well starting from 100–150 features in the image using our chosen features and detectors. This threshold of 150 detected features was reached in 20.7% of all frames with active indicator, and in 8.1% with inactive indicator. Thus, the indicator significantly increased the probability for successful relocalization, which confirms **H3**. While those ratios may in overall appear low, it has to be kept in mind that in practice, a certain amount of frames will always be subject to motion blur, and 20% of frames with sufficient features still yields on average five frames per second (at a video frame rate of 25 frames per second), which is sufficient for continuous visual localization. Figure 5.14 illustrates, based on an exemplary excerpt of the experiment's data, how the number of features per frame was correlated with the phone inclination and the state of the indicator.

The experiment also unveiled a clear preference for carrying poses. All subjects carried the phone in a way that VR was activated, and only raised the phone to the AR position when told so by the visualization. Soon after the indicator disappeared, they returned to the more comfortable VR carrying position. Subjects responded that they found the pose-dependent switch between AR and VR convenient (MD = 2.5). They also understood the meaning of the indicator: They agreed with MD = 3 to the statement *"What I should do when the indicator appeared was clear to me"*, and with MD = 3 to the statement *"I have been motivated by the indicator to raise the phone up"*.

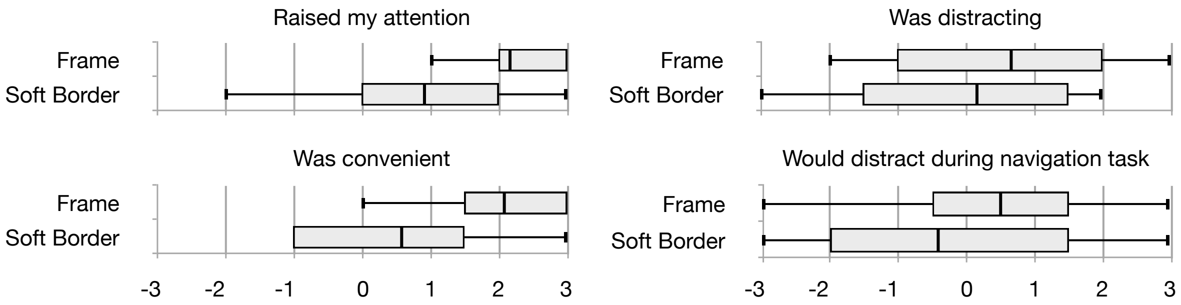


Figure 5.15: User feedback on *Frame* and *Soft Highlight* object visualization. Answers were given on a 7-point Likert scale, ranging from -3 (strongly disagree) to +3 (strongly agree).

H4: Object Highlighting Methods

On a Likert scale from -3 to +3, subjects indicated that *Frame* drew significantly more attention to the poster (2.2, SD = 1.5) than *Soft Highlight* (0.9, SD = 1.6), $p < 0.05$. Given that the visualization signals a possibility to interact with the object, they found *Frame* more convenient (2.1, SD = 1.0) than *Soft Highlight* (0.6, SD = 1.5), which was likewise a significant effect ($p < 0.05$). One reason might be that the semi-transparency of *Soft Highlight* complicated readability of text on the poster. Regarding distraction, we measured no significant difference between *Frame* (0.5, SD = 1.7) and *Soft Highlight* (0.0, SD = 2.0). The results are summarized in Figure 5.15.

5.3.4 Discussion and Lessons Learned

We now distill lessons learned from a comprehensive view of the findings of both the online and the experimental study. We also include issues that have not been explicitly addressed in our presentation of results, but which have become evident in the course of our studies, or were mentioned by participants when “thinking aloud”.

VR as Main Visualization

The experiments confirmed the advantages of VR unveiled in the online study, and showed additional ones. Taking these findings together, we recommend VR as main visualization for indoor navigation systems, for the following main reasons:

- **Effectiveness**: The perceived correctness of instructions is higher.
- **Efficiency**: The destination is reached faster.
- **Convenience**: The visualization is more popular and perceived as more convenient.

Convenience needs further explanation here. First, VR did not require to carry the phone upright as AR does, so it was perceived more comfortable. This issue of discomfort was not only raised by subjects in the AR condition. It was also confirmed by the second experiment, where subjects could choose freely how to carry their phone, and visualizations were set automatically according to the pose. Almost everyone “automatically” chose VR. Third, multiple subjects reported that they felt uneasy in AR mode because other people might think that they would take photos or videos of them. These findings related to convenience have probably been the decisive factor for the final user preference for VR. However, this does not mean that AR is entirely useless, which we elaborate in the following.

Combination of VR and AR

AR has shown its strength in two cases. First, it can help acquiring good query images using the feature indicator. The *Bubble Level* visualization contributed to a rise of visual features in query images, thus increasing the probability of reliable relocalization – but this visualization is only reasonable to use in combination with AR. Second, AR conveniently integrates object highlighting. By enabling AR-based object interaction, users can possibly be intrinsically motivated to target these objects with the camera, and again, this potentially leads to better query images.

What we essentially learn is that a good user interface could benefit from a combination of both AR and VR. The visualizations could be chosen automatically, based on the scheme displayed in Figure 5.16.

- The choice of AR or VR is dependent on the device's pose: When holding the phone in a way that the camera points towards the floor, as depicted in Figure 5.1b, AR makes little sense, no good query images can be captured and, moreover, AR guidance instructions cannot be optimally aligned with the environment. Instead, the user can then compare the panoramic image on the phone with her field of view. By contrast, AR only is useful when the phone is held upright (see Figure 5.1a) and she can see the environment "through" the phone. In this mode, objects for interaction can be found with the highlighting function. Thereby, the localization quality is implicitly improved, as continuously new material for image matching is collected.

- The default visualization is VR, as the device will most likely held facing the floor. In this mode, the certainty of the location estimate will expectably decrease with time, as distinctive images are less likely to be found in this pose. However, as proven in our two conducted studies, the VR interface can deal with inaccuracies in satisfactory manner. For still being able to update the location estimate as the user moves, the device must use (less accurate) sensor-based dead reckoning or other relative positioning techniques. One promising alternative is *visual odometry*, which we have investigated in earlier research [128, 316].

- When the location estimate becomes insufficient even for the robust VR mode, the system could switch to AR and ask for relocalization with the help of the *Bubble Level* indicator. The AR mode is in this case unavoidable and enforced, no matter in which pose the user holds the phone. As soon as the visual relocalization was successful, the user can again choose pose-dependent between AR and VR.

Finally, we want to give one more argument for why we do not postulate using either AR or VR alone: The considerable standard deviations (see the whiskers in Figure 5.13, in particular for the AR ratings) reflect that users are heterogeneous and do not perceive the same. This is an additional motivation for letting users choose between alternative visualizations.

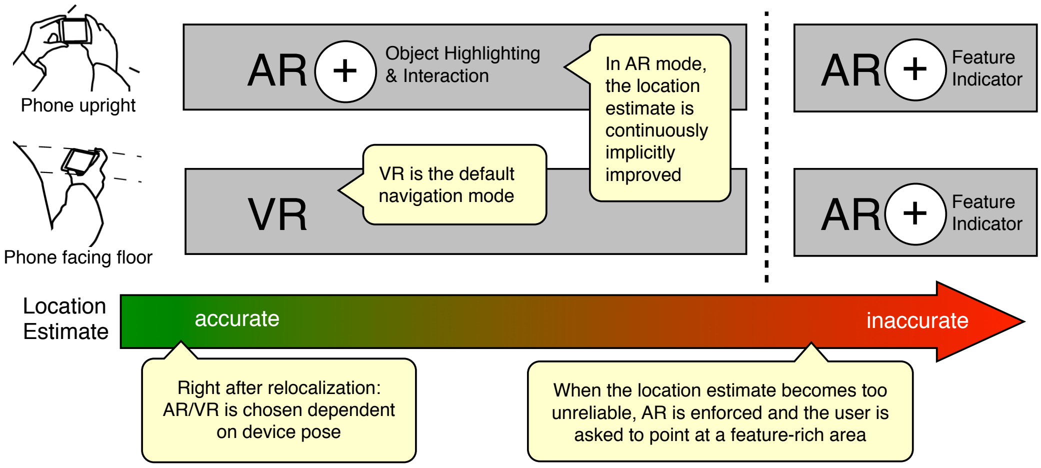

Phone upright
Phone facing floor
AR + Object Highlighting & Interaction
In AR mode, the location estimate is continuously implicitly improved
VR
VR is the default navigation mode
AR + Feature Indicator
AR + Feature Indicator
Location Estimate
accurate
inaccurate
Right after relocalization: AR/VR is chosen dependent on device pose
When the location estimate becomes too unreliable, AR is enforced and the user is asked to point at a feature-rich area

Figure 5.16: One suggestion for the interplay of augmented reality and virtual reality visualizations dependent on how the user carries the phone and on the accuracy of the location estimate.

How to Improve the VR Visualization

The user studies revealed some problematic aspects with the VR visualization. The first two of them can be summarized as "discrepancies between the real and the virtual world".

First, exposure and lighting conditions were not the same between all panorama images used for the guidance instructions. This did not impact route finding, but was nevertheless noticed by subjects, and contributed to irritations – especially when panoramas updated in one-second intervals. Automatic post-processing [324] could partly, but probably not always satisfactorily improve the quality of recorded reference images. Particularly, it has to be considered that an indoor navigation system might include also crowd-based reference images taken by users, which then presumably are of varying quality (with regard to technical aspects, such as camera quality, as well as to content, e.g., people occluding distinctive scene elements).

Second, public environments are subject to change (consider, e.g., renewed advertisements or posters, updated shop window decorations, etc.). As a consequence, reference images become outdated and need to be updated as well. This could, e.g., be done by crowd-based updates (query images taken by users are used as new reference images). The details of such a collaborative update mechanism is not in the scope of this thesis, but there is another aspect to outdated image material: Since humans use landmarks for orientation, it can be irritating if a significant object shown in the panorama is not present any more or looks different in the real world.

A third issue is the update frequency of panoramas. In our experiments, the panorama view was updated by the "wizard" every few meters. This corresponds to the distance when typically a new location estimate would be available in a live localization system. We learned that this frequency is not optimal. If the shown locations were incorrect (in the low-accuracy condition), the panorama seemed to rapidly "jump" from one location to another. Subjects reported that they were irritated when the screen content refreshed so frequently, especially when not permanently looking at the display. Moreover, panoramas were slightly different in perspective and lighting (as discussed above), which entailed that subjects had to "re-check" more often their position in reference to the panorama each time they looked back at the display. Some stated to have looked at panoramas only when they approached turn locations, i.e., decision points.

This brings us to the idea of varying the frequency in which panoramas are updated during a path. Instead of showing always the closest available reference image to the current location estimate, a characteristic subset of panoramas could be used for guidance along the route, illustrating particularly the turns and difficult parts. This could reduce the cognitive effort required for visually matching panoramas with the real world, at similar quality of guidance. We will further investigate this idea and criteria for defining such a subset of images in the following section.

Summing up, we have learned how users can benefit from a synthesis of AR and VR interfaces. On the one hand, we have found that especially VR complements well visual indoor localization systems, providing a faster and more reliable navigation. On the other hand, we also have identified issues of VR that can be improved. This motivated us to one more design iteration where we focus on VR, and further optimize this visualization.

5.4 Navigating Using Decision Points

From the previous two studies, we learned that VR-based instructions were beneficial, but also overwhelming with relation to their update frequency, and that irritations could occur due to inconsistent and outdated reference images.

To overcome these shortcomings, further pursuing our iterative design approach, we suggest a novel concept which we call decision-point-based navigation (DPBN), an extension of our previous VR concept. The basic idea is here that the system does not show a new panorama with each localization update (i.e., every few meters), but only if the user has to make a decision at this location (turn left, right, use the stairs, etc.). This brings us the following advantages:

- The interface must be updated less frequently, adding to a more calm, thus less irritating, visualization. If there are exposure changes between multiple subsequent panoramas, it is expected that a potential disturbing effect decreases, as there are longer intervals between subsequently shown panoramas.
- The likelihood that the user sees an outdated panorama is reduced, as only a fraction of available panorama photos lying on the path is actually presented to the user. Real-world changes that render reference images outdated are likely to occur for *landmarks* (i.e., salient objects). However, decision-point-based navigation differs from the known concept of landmarks, as landmarks not necessarily occur only at decision points [272], and sometimes not even lie on the route. We emphasize that decision points in our system do not need to be prominent points (as landmarks are), but simply provide a visual impression of the location where the decisive action has to be made.
- DPBN is more error-tolerant than the previously presented VR mode. Let d be the distance between two subsequent decision points (empirically determined as typically

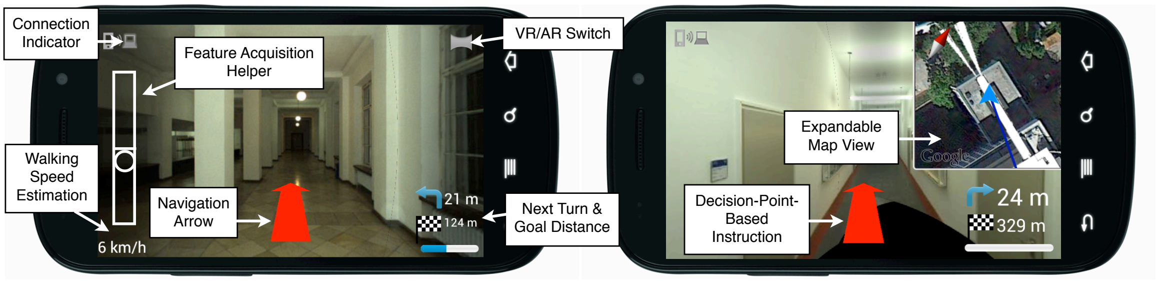


Figure 5.17: Iterations of the prototype. Left: interface based on virtual reality (VR) using panorama images of the current location. Right: Using panoramas only of decision points (DPBN) and an additional map view for overall orientation.

ranging between a few meters and several dozens of meters in large buildings). The shown panorama is then correct even when the localization uncertainty is below $d/2$. In the conventional VR approach, the visualization would already be erroneous when there exists an image closer to the actual location than the image that is shown (and this would actually be noticed as an error by the user!).

- DPBN conforms to the mental model of human route memorization. Humans would describe a route like *"turn right in the hall when you face the library, then walk straight ahead until the elevator, there turn left..."*. The sequence of decision point panorama can thus be considered as a visual route summary.

5.4.1 Research Questions

We investigated the following research questions (RQ) for the newly introduced DPBN approach.

RQ1 Does DPBN have an effect on efficiency?

Are users as fast as with continuous panoramas, or do they need more time to reach their destination when the interface only shows them decision points?

RQ2 Is DPBN as convenient as continuous panoramas?

Besides the quantitative comparison, we investigate which mode users prefer and how well they feel guided in either DPBN or the continuous mode.

RQ3 What usage patterns can be identified?

Furthermore, we are interested in observing usage patterns and strategies with panorama-based navigation. We therefore let subjects use the system in offline mode to see what we could learn for designing an ideal route description.

5.4.2 Experimental Evaluation

Implementation

We extended the prototype used in the previous experiment to support three versions of the VR mode: In *DPBN-auto*, panoramas of decision points are shown and updated automatically as the user is walking. As soon as the user has passed a decision point, the next decision point is shown. *DPBN-manual* contains the same list of decision points, but they are not updated automatically. Instead, the user can flick through the sequence of images manually step by step. This mode supports navigation even when localization temporarily fails, because the user can match the images with the real environment, and swipe to the next instruction once she passed a decision point. The third mode is *Continuous*, which is identical to the VR mode described in the previous study. The modes were implemented as WOz system, similar to the previous study.

AR mode was not included in this version, as we wanted to compare DPBN to the conventional VR mode. Nevertheless, we added a “relocalization” function: In any of the three modes, users have the possibility to retrieve the closest panorama to their actual location. For maximal realism, the trigger for the relocalization is raising the phone to an upright pose (as if taking a photo). This corresponds to the pose in which relocalization would be most likely successful in a real system. The pose was detected by the phone’s IMU. To reduce unnecessary complexity, the feature acquisition procedure with the *Bubble Level* indicator was not simulated; instead, the correct panorama was immediately loaded, and the system continued to work in the respective condition (*DPBN-auto*, *DPBN-manual*, or *Continuous*).

Method

We conducted a within-subjects experiment with three versions of the prototype: continuous panoramas (in the following referred to as *Continuous*), automatic decision points (*DPBN-auto*) and manual decision points (*DPBN-manual*). In each condition, subjects had to navigate on a different path inside the TUM university campus. The conditions were counterbalanced using a Latin square design. The three paths had a lengths of 332, 220 and 316 meters and contained 7, 11 and 7 decision points. The varying length and complexity of the paths allowed us to investigate how users behave on long segments without decision points, as well as in sequences with rapidly following instructions with many possibilities to choose a wrong way. Subjects were instructed to rely on the instructions given by the application. In *DPBN-manual*, they could swipe back and forth between panoramas as they wanted. They were also free to use the relocalization feature. The experimenter walked closely behind the subject and sent the panorama images to the subject’s device using the WOz application (see Figure 5.11a).

In all conditions, we measured the time until the destination was reached. In *DPBN-manual*, we logged all user interactions on the smartphone and recorded when a location update was received. By this, we were afterwards able to compare the decision point the user currently looked at with the “correct” next decision point. This helped to the identification of “strategies” when dealing with panoramas in manual mode. At the end of the experiment, we collected subjective data with a questionnaire.

Participants

12 participants (3 females, 9 males) took part in this initial DPBN study (a follow-up study with 18 subjects was conducted afterwards; see later in this section). The average age was 25 (SD = 3). Nine subjects were rather experienced with smartphone usage, but not with indoor navigation (only one indicated to have used a respective system before). None of them were familiar with the university building where the study took place.

Results

Efficiency (RQ1)

Subjects took on average 196 s (SD = 19.1) to the destination in *Continuous*, 208 s (SD = 51.6) in *DPBN-auto* and 263 s (SD = 65.9) in *DPBN-manual*. Results are visualized in Figure 5.18a. Measurements in all conditions were normally distributed ($p > 0.05$ in a Kolmogorov-Smirnov test [159]). With a one-way repeated-measures ANOVA, we found a significant effect of conditions on task time ($F(2, 22) = 9.85$, $p < 0.001$, partial $\eta^2 = 0.28$). Post-hoc t-tests with Bonferroni correction showed no significant difference between *Continuous* and *DPBN-auto* ($p > 0.05$), but between the other conditions with $p < 0.05$. *DPBN-auto* is hence essentially as efficient as the *Continuous* mode. By contrast, subjects needed significantly more time in *DPBN-manual*.

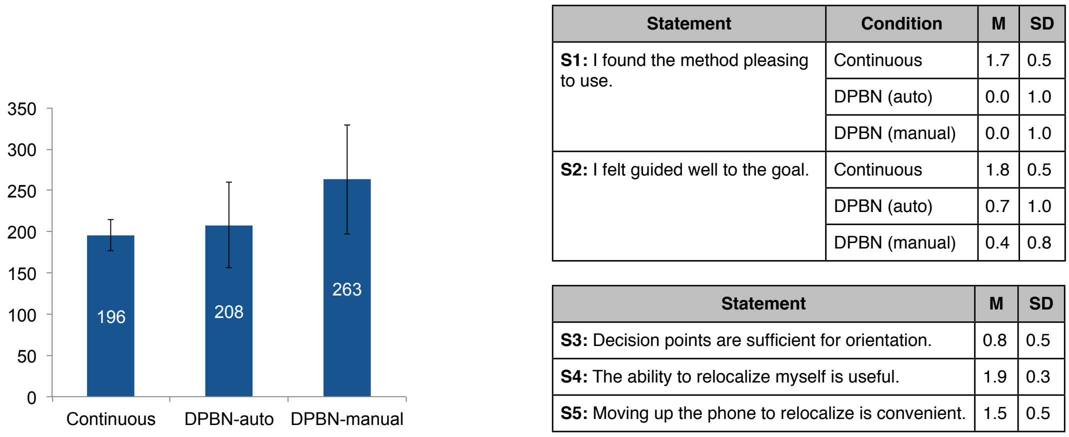
User Preferences (RQ2)

In questionnaires after the experiments, subjects gave feedback to five statements *S1* to *S5* (see the tables in Figure 5.18b). Results are indicated on Likert scales ranging from -2 (strongly disagree) to 2 (strongly agree).

S1 addresses the question how pleasant to use users found the different methods. A Friedman test showed a significant effect between methods ($\chi^2 = 13.68$, df = 2, $p = 0.001$). Participants rated the *Continuous* mode significantly more pleasing to use than both DPBN conditions (*S1*, 1.7 vs. 0.0, $p < 0.005$ in a post-hoc Wilcoxon signed-rank tests with Bonferroni correction).

Likewise, there was a significant effect on the perceived guidance quality ($\chi^2 = 11.49$, df = 2, $p = 0.003$). Subjects felt to be guided better to the destination (*S2*) in continuous mode (1.8) than in DPBN (0.7 in automatic, 0.4 in manual mode). These effects were significant between *Continuous* and *DBPN-auto* ($p < 0.05$), and highly significant between *Continuous* and *DBPN-manual* ($p < 0.001$) in post-hoc Wilcoxon signed rank tests with Bonferroni correction.

While those results imply that subjects were less satisfied with DPBN, there is above-average agreement (0.8, SD = 0.5) that decision points are sufficient for orientation (*S3*). This is a hint that the DPBN principle essentially works (as confirmed by the results for RQ1), but received less acceptance with subjects. In order to find out how acceptance could be further increased, we will take the observed user behavior in manual mode (RQ3) into account.



Statement	Condition	M	SD
S1: I found the method pleasing to use.	Continuous	1.7	0.5
	DPBN (auto)	0.0	1.0
	DPBN (manual)	0.0	1.0
S2: I felt guided well to the goal.	Continuous	1.8	0.5
	DPBN (auto)	0.7	1.0
	DPBN (manual)	0.4	0.8

Statement	M	SD
S3: Decision points are sufficient for orientation.	0.8	0.5
S4: The ability to relocalize myself is useful.	1.9	0.3
S5: Moving up the phone to relocalize is convenient.	1.5	0.5

(a) Average time per condition to reach the destination in the study using our navigation system prototype. The error bars indicate standard deviations.

(b) Qualitative feedback on the system. The mean agreement (M) to statements S1 to S5 is indicated on a 5-point Likert scale (1 = strongly disagree, 5 = strongly agree). SD indicates standard deviations.

Figure 5.18: Evaluation results of decision-point-based navigation (DPBN) versus the continuous VR mode, as described in earlier sections.

User Strategies (RQ3)

We identified two major strategies of user behavior in the *DBPN-manual* condition. One group of users always displayed the decision point lying ahead, walked until the shown location was reached, and swiped then to the next decision point. These subjects almost never made use of the relocalization feature. Another group did not wait until the next decision point was reached. Instead, they continuously used the relocalization feature to see a panorama matching as close as possible with the current position. In fact, they relocalized so frequently that the effect was almost similar to the *Continuous* condition. However, no matter which of these strategies users applied, they found the relocalization feature (*S4*) extremely useful (1.9, SD = 0.3).

A possible explanation for frequent self-relocalization is that subjects were unconfident especially on long route segments. In that case they desired an intermediate "control" point to confirm that they are on the right way. In *DPBN-auto*, on average 10.3 locations were shown per path; in *Continuous* it were 52.3 decision points. In the *DPBN-manual* condition, subjects on average swiped 15.3 times to a new decision point and used 9.3 times the relocalization function.

Implications

These results have the following implications:

- Subjects reached the destination with *DPBN-auto* as fast as with continuous panoramas (the time difference was not statistically significant). We can therefore draw the conclusion that it does not affect performance if we reduce the panorama update frequency from *Continuous* mode to a lower frequency, and only show decision points.

- Subjects prefer *Continuous* mode over *DPBN-auto*, possibly because the number of decision points was too low in *DPBN-auto*. Yet, looking at the sum of swipes and relocalizations in *DPBN-manual*, the high number of panoramas of the continuous mode is not necessary either. A compromise between the two could be the optimal solution.
- We need to investigate where the additional instructions in *DPBN-auto* shall be inserted, and more generally, what are criteria for good decision points. One example could be to insert intermediate panoramas on long route segments without decision points. When there are many options to leave the straight path, confirmations to stay on the route could add confidence.
- To make it easier to detect whether a decision point is reached, a distance estimation until the currently displayed panorama could be shown. Already passed decision points could be marked so that users can see at a glance which part of the route has already been completed in the list of panoramas.
- When the user is walking fast, signifying she is sure about her way, DPBN could be used. As she slows down, e.g., in case of uncertainty (detected through the phone's accelerometer), the system could switch automatically into the continuous mode to give more hints for orientation.

Follow-up Study

We conducted a follow-up study to investigate the open issues outlined above. We strived to increase users' confidence when navigating with the application by two measures. First, we added intermediate panoramas in long route segments in the *DPBN-auto* mode, so that there were now in total 11, 14 and 10 panoramas on the three paths (instead of previously 7, 11 and 7). Second, we added a map view (see right screenshot in Figure 5.17), which is shown in a corner of the screen and which can be enlarged to full-screen. The map view is intended to provide better overall route awareness and to increase familiarity with the interface, as maps are well-known from standard navigation applications.

With these enhancements, we formulate the following additional research questions.

RQ4 How does the enhanced DPBN compare to the previous version in terms of user acceptance?

RQ5 What is the acceptance of the map view, compared to DPBN?

RQ6 What are criteria for good decision points?

Method

With the updated prototype, we conducted a within-subjects experiment with three versions of the prototype: continuous panoramas (in the following referred to as *Continuous*), automatic decision points (*DPBN-auto*) and manual mode (*Manual*). *Continuous* and *DPBN-auto* were implemented in the same way as in the previous study. In manual mode, we made a modification: While subjects in the previous study had a list of decision points available, they now did not get any panorama at all, as long as they did not explicitly request one. If they did so, the closest panorama to the current location was shown. This allows us to investigate at which locations subjects potentially want to receive instructions. In *Manual*, subjects were instructed to request an instruction only when they otherwise would be unsure about their way. All other parameters (paths, instructions and the WOz setup) were the same as in the previous study. After the experiments, subjects answered a questionnaire.

Participants

18 participants (6 females, 12 males) took part in this study, the average age was 32 years (SD = 13). None of them had participated in the previous experiment. 12 subjects own a smartphone, but do not use navigation applications overly often (with an average of 0.2 on a 5-point Likert scale, where -2 = never and 2 = frequently; SD = 1.4). One subject stated to have used indoor navigation previously.

Results of the Follow-up Study

RQ4: Effect of the Enhanced DBPN

Subjects indicated that *DBPN-auto* was pleasant to use (1.7, SD = 2.0); for *Continuous* the agreement was 0.9 (SD = 1.0). This is a significant difference according to a Mann-Whitney U-test [198] (W = 90.5, Z = -2.52, $p < 0.05$). This result is opposed to the previous study, where *Continuous* was rated significantly better – obviously, the changes made to DPBN in this iteration could in fact make DPBN not only superior in terms of efficiency, but also user preferences.

In direct comparison with the previous study, the convenience of *DBPN-auto* (agreement to the statement *"The system is pleasant to use"*) significantly increased from MD = 0 to MD = 2, according to a Mann-Whitney U-test (W = 19, Z = -4.03, $p < 0.05$). The average convenience of *Continuous* decreased from MD = 2 to MD = 1, which is likewise a significant effect (W = 152, Z = 2.02, $p < 0.05$).

With the enhanced DPBN, subjects also feel significantly better guided to the goal (MD = 2) than in the previous study (MD = 0.5) according to a Mann-Whitney U-test (W = 36, Z = -3.47, $p < 0.05$). There was no significant difference for the *Continuous* mode relating to that aspect. The results support the assumption that we have reached our goal and the enhancements made to DPBN made users feel more comfortable with it, so that they favor it over continuous navigation.

RQ5: Usage of Map versus DPBN

11 of 18 subjects (61%) indicated to have relied mostly or slightly more on the panorama view, compared to maps. Only 5 of 18 (28%) preferred the map view. Two subjects used both interfaces equally (see Figure 5.19). In almost half of all runs (17 of 36), subjects stated that they did not look at the map interface at all. While the map view was generally rated as helpful (averagely 1.3 on a Likert scale ranging from -2 to 2, SD = 1.0), there was only average agreement that the map view is sufficient (0.1, SD = 1.0). Given that maps are the prevalent known visualization on mobile navigation systems, this is an interesting result. We see it as an indicator that subjects were open-minded towards our proposed VR-based decision point visualization and adopted it very well.

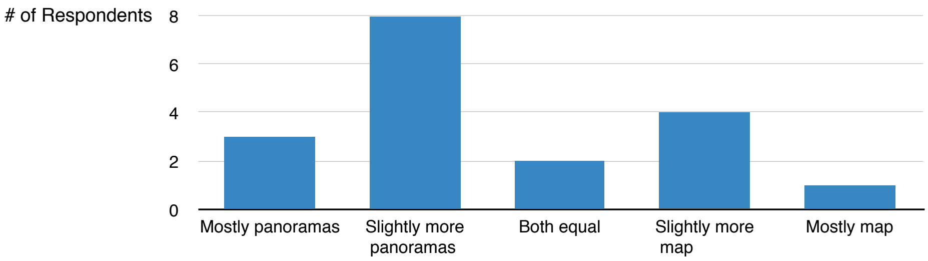


Figure 5.19: Histogram of indicated panorama versus map usage. Subjects tend to rely more on the virtual reality panorama views during navigation.

RQ6: Criteria for Good Decision Points

In order to assess how well our predefined decision points in DPBN match users' expectations, we compared the predefined locations for decision points in the *DPBN-auto* condition with the locations where subjects requested decision points themselves in the *Manual* condition. We use the term request location (RL) for the location of the decision point (DP) in our database lying closest to the location where the user triggered the request. We interpret a request as a *match* if the distance d(RL, DP) lies below a certain threshold t:

$$\sqrt{(RL.x - DP.x)^2 + (RL.y - DP.y)^2} < t$$

For t = 5 meters, we observe that 74% lie below the threshold, while for t = 7 meters, the match rate is already 88%. 90% of all requests are closer than eight meters from the original decision points. These results indicate that the decision point locations in this second iteration of DPBN were already very close to users' expectations. This possibly explains the fact that *DPBN-auto* mode was now preferred over *Continuous* mode (unlike in the first iteration of the decision-point-based system).

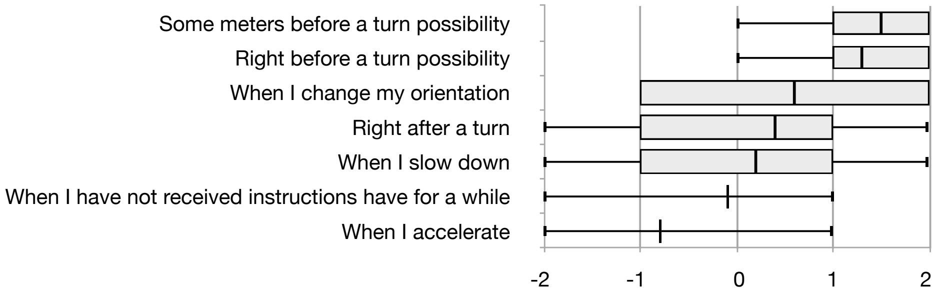


Figure 5.20: Importance of new instruction (decision points) at certain locations or situations. Answers given on a 5-point Likert scale from -2 = strongly disagree to 2 = strongly agree.

To further inform the choice of good decision points, we asked subjects at which locations or in which situations they consider instructions especially important. Figure 5.20 lists the results on a Likert scale, where -2 corresponds to strong disagreement and 2 to strong agreement with a location or situation. According to the answers, most important are instructions some meters before the possibility to choose an alternative path (1.5, SD = 0.6) and right before such a point (1.3, SD = 0.8). Another situation where instructions were preferred was when users change their orientation (0.6, SD = 1.2). Turning around and looking backward could be an indicator for disorientation, so that they need a cue to follow the right path. The demand for instructions when slowing down (0.2, SD = 1.1) and right after a turn (0.4, SD = 1.2) was slightly above average. While slowing down could, similar to turning around, indicate uncertainty, an instruction after a turn would rather have a confirmative effect (*"your turn was correct, now follow this path"*). A neutral result was yielded for the question whether instructions should be shown time-based, i.e., when no instructions have been received for a while. When accelerating, subjects clearly did not require instructions (-0.8, SD = 0.7). This is reasonable, as they are more certain where to go in that case, and might not look at the display when walking fast.

5.4.3 Discussion and Lessons Learned

In this section, we sum up the design approach we have pursued with the goal to develop a user-friendly and efficient multimodal interface for indoor navigation. We have performed an iterative development process that can be summarized into four stages, each consisting of a design and evaluation phase (see a summary in Table 5.1).

Visualization Concept	**Subject of Investigation**	**Type of Evaluation**	**Number of Participants**
VR/AR-based, 4 feature indicators, object highlighting	Design mockup	Online	81
VR/AR-based, 1 feature indicator, object highlighting	Prototype	Experimental	12
DPBN as VR extension	Prototype	Experimental	12
DPBN refinement, map view	Prototype	Experimental	18

Table 5.1: Overview of the iterative research approach pursued in this chapter.

In an iterative design approach, we have refined the VR-based approach towards a decision-point-based approach (DPBN). We have shown that DPBN is equal to conventional VR in terms of navigation time, i.e., the reduced number of instructions does not negatively affect efficiency. In addition, DPBN is more robust against inaccuracy since determining the next decision point with relation to the current position requires a lower localization accuracy, compared to displaying always the matching panorama according to the current position.

With a second iteration of DPBN, we improved the placement of decision points, addressing the issue that subjects sometimes felt lost when there were too little decision points. We found that it can be beneficial when instructions are shown already several meters before a turn, instead of immediately before that turn. These improvements addressed problems in the first DPBN version that was less popular with users compared to conventional VR. With the improved version, subjects favored the DPBN version over the conventional VR interface, so that DPBN can be seen not only equal to VR in terms of efficiency, but even superior in terms of user satisfaction.

Limitations of the Studies in this Chapter

We acknowledge that the experimental evaluations presented in the previous sections have limitations. First, we used simulated localization data for all experiments. Controlling the output of the localization system allowed us to isolate the *effectivity and efficiency of the user interface* as variable of investigation. However, it has to be kept in mind that the UI responses were generated by the WOz technique, and that a self-contained system might yield different results. We also want to note that it was not the goal of the conducted research to evaluate the accuracy of a visual localization system. While we argue that visual localization is a promising approach for indoor navigation (see Section 2.2.4), an investigation whether visual localization is superior to other localization techniques was not a research question and exceeds the scope of this dissertation.

5.5 Summary and Lessons Learned

We have reported on the design and implications of multiple user interface iterations for optimizing the indoor navigation task. Here, at the end of this chapter, let us discuss what the previously presented findings mean for the domain of indoor navigation with relation to multimodality.

Our work includes multiple (independent and parallel) modalities on different levels. The localization method, at a low level, relies on the visual modality, which is already relatively new [302]. Further, the *Device Gesture* modality is used to automatically switch between different visualizations (AR and VR) that are best suited for the respective poses. In Section 5.3.4, summarized in Figure 5.16, we have made a proposal for how both visualizations could fluently be combined. This suggestion shows how indoor navigation applications could benefit from multimodal interface approaches. The feature indicator contributes to a win-win situation for both the user and the system. Not only is the user experience improved, as route guidance is better in case of inaccuracy. Also the localization certainty is improved, as the feature indicator UI element encourages the user to record better query images. An important lesson learned is here that the MUSED interfaces must be compliant with the user's mental model [122, pp. 49ff.]. This win-win situation was only possible because users intuitively raise the phone when they want to use the AR interface. This shows the importance of a context-aware choice of modalities. In Chapter 6, we will present a solution for such context-driven modality choices with a software framework.

User satisfaction results for the novel multimodal interfaces were highly encouraging. In the final study, we saw that the VR-based DPBN interface was used more than the map interface, which people are familiar with. The research presented in this chapter is just an example of how multimodality can be successfully used in the area of indoor navigation. In the following, we just outline some possible research directions for even further enhancing the interface as it is now. In that sense, we want to motivate researchers to experiment with and try out new interaction paradigms, without being afraid that people always just use what they know.

- We have seen in the answers of the final study that walking speed is an indicator for the certainty of the user (as users were interested in receiving additional decision points). When the route gets complicated or instructions are insufficient, users may likely slow down. Hence, walking speed estimation could be integrated as a cue for triggering additional instructions and for modifying the UI in other dimensions, e.g., showing textual instructions or additional landmarks.

- We have, in our research with MUSED interaction, focused on sensors (camera, accelerometer, compass) and input modalities built on top of them. The proposed MUSED interaction concepts are not limited to the exemplarily used sensors and modalities we have presented here. Further research could focus stronger on output modalities as well. For example, the haptic channel can be included to inform the user on new instructions when the user does not look at the mobile device or when it is, e.g., in the pocket. Speech is an option for the output channel as well.

- Personal preferences can inform route guidance. This does not only entail that accessibility requirements are considered (e.g., avoiding stairs and using the elevator instead), but also that, e.g., badly illuminated corridors are avoided at evening hours, even if the alternative way is longer.

- Route guidance can adapt to contextual factors, like crowdedness in a certain area of the building. Imagine a museum scenario where the application's tour suggestions contain routes that are currently less frequented. By managing the stream of visitors like this, visitors could have a better view of the exhibits and a better overall experience.

- Cooperative localization and exchange of the position using device-to-device communication could further leverage the accuracy of indoor localization. A device with a better location estimate could share its information via Bluetooth Low Energy (BLE) with other users nearby (cf. low-distance beaconing as introduced in iOS 8[48] or Android L). This approach is, e.g., used in the automotive domain [283].

- By combining of indoor and outdoor navigation, seamless mobility chains could be created. A calculated route could then integrate, e.g., walking segments (pedestrian navigation), self-driven segments (car navigation), and train/bus segments (public transport integration), where the handheld device switches between different UIs and modalities for instruction presentation. As an example for a platform integrating diverse mobility providers, see [80].

Part III

Design and Evaluation of Multimodal Applications

Chapter 6

Designing and Implementing Mobile Multimodal Systems

6.1 Problem Statement and Research Questions

After we have presented and investigated selected examples for multimodal systems in detail, we are now approaching in a systematic way how MUSED applications can be created. The implementation of novel interaction modalities and sensor-driven behavior can be a tremendous effort, often involving recurring tasks. As a solution to speed up and simplify MUSED application development, we present a software framework whose feature set is informed by a (questionnaire- and focus-group-based) requirements analysis. In this chapter, we describe this framework in detail and present sample applications developed with the framework to highlight its potential. By several implemented use cases we demonstrate the available design space, also with reference to research apps presented in prior chapters.

As a second focus of this chapter, we take up a user-centered approach to further inform the design of multimodal systems. First of all, we investigate current modality usage habits, e.g., which modalities users consider useful in different situations. We then conceive and evaluate user interfaces for defining and achieving awareness of multimodal behavior.

This chapter answers the following high-level research questions:

- How can the implementation of multimodal systems be simplified?
- How can users be supported in adapting the multimodal behavior of mobile systems to their needs, and how can awareness on currently active modalities be achieved?

This chapter is partly based on two papers we have published in 2014 [222, 223].

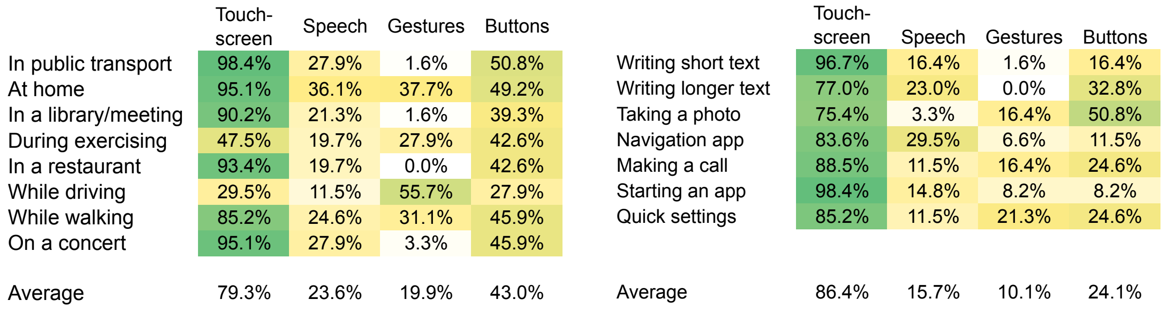
6.2 Elicitation of Requirements

6.2.1 Current Use of Multimodality – Two First Surveys

For initial insights which modalities are currently predominantly used on mobile devices in everyday life, and under which circumstances, we report on two surveys we conducted prior to the development of the framework and the user interfaces presented later in this chapter.

Usage of Modalities

We begin with large-scale results on situation- and task-specific modality usage. The data stem from an online survey conducted with 61 participants (26 females, 35 males), aged between 16 and 36 years (M = 21, SD = 4). The participants were asked to indicate which modalities they use in particular situations or locations (e.g., at home, while walking, driving or in public transport, in a public space like a restaurant, etc.). In addition, they were asked to report on their modality usage with relation to specific tasks, e.g., entering text, making a call, or taking a photo. Multiple answers were allowed.



	Touch-screen	Speech	Gestures	Buttons
In public transport	98.4%	27.9%	1.6%	50.8%
At home	95.1%	36.1%	37.7%	49.2%
In a library/meeting	90.2%	21.3%	1.6%	39.3%
During exercising	47.5%	19.7%	27.9%	42.6%
In a restaurant	93.4%	19.7%	0.0%	42.6%
While driving	29.5%	11.5%	55.7%	27.9%
While walking	85.2%	24.6%	31.1%	45.9%
On a concert	95.1%	27.9%	3.3%	45.9%
Average	79.3%	23.6%	19.9%	43.0%

(a) Situation-specific usage

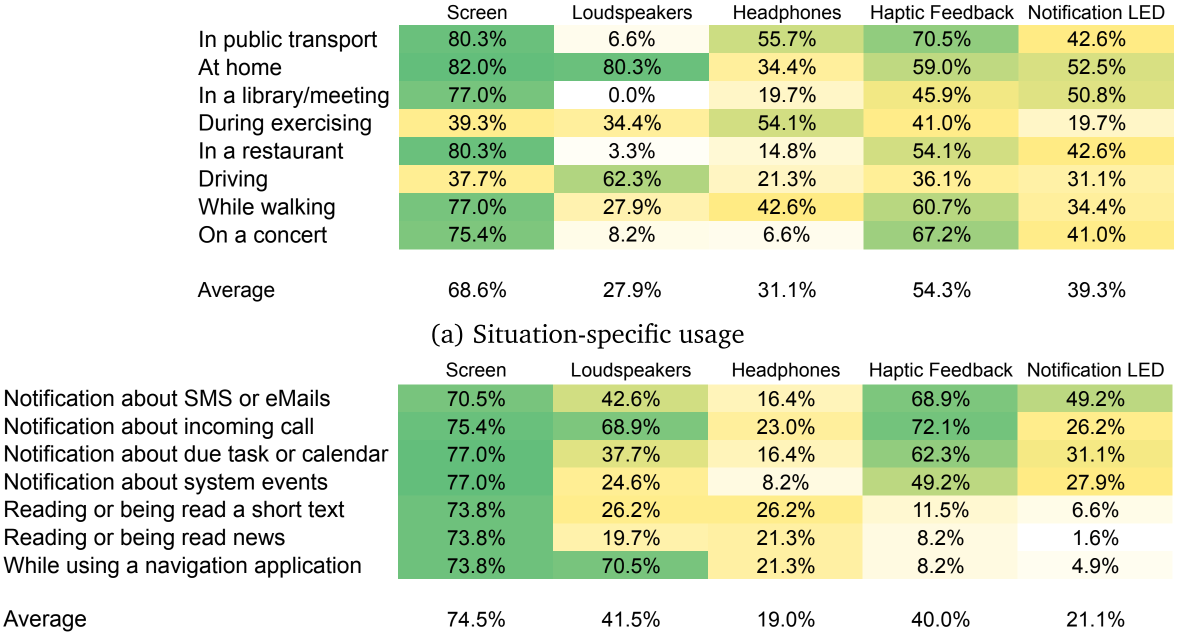
	Touch-screen	Speech	Gestures	Buttons
Writing short text	96.7%	16.4%	1.6%	16.4%
Writing longer text	77.0%	23.0%	0.0%	32.8%
Taking a photo	75.4%	3.3%	16.4%	50.8%
Navigation app	83.6%	29.5%	6.6%	11.5%
Making a call	88.5%	11.5%	16.4%	24.6%
Starting an app	98.4%	14.8%	8.2%	8.2%
Quick settings	85.2%	11.5%	21.3%	24.6%
Average	86.4%	15.7%	10.1%	24.1%

(b) Task-specific usage

Figure 6.1: Input modality usage in different situations and for different applications and use cases. The diagrams indicate percentages of agreement (white: low agreement, green: high agreement).

Figure 6.1 summarizes the results for input modalities. We see that both for situation-specific and for task-specific usage, the touchscreen is the prevalent input method. For almost all situations and tasks, 75% to almost 100% of participants confirmed to prefer this modality, followed by hardware buttons as second-most answer. However, there are exceptions: While driving or exercising, the touchscreen is only used by 29.5% and 47.5%, respectively, and gestures become more important. When looking at tasks, modalities other than touchscreen input can be observed for taking a photo (50.8% use a hardware button), writing longer texts (32.8%, probably corresponding to devices with hardware keyboards), quick settings (21.3% activate settings by gestures), and navigation applications (29.5% use speech input to enter their destination).

The results for output modalities are shown in Figure 6.2. They exhibit larger divergence than for input: Although the screen is still the prevalent way to receive information, subjects likewise use sound (via loudspeakers or headphones), haptic feedback, or visual notifications using a LED. The usage of these channels differs again by situation. Loudspeakers are mostly used at home (80.3%) and in the car (62.3%). In public transport or during exercising, subjects prefer headphones (55.7% and 54.1%, respectively). Haptic feedback (i.e., vibration) shows generally higher usage percentages than aural notifications. It is particularly used in loud environments (e.g., in public transport with 70.5%, or on concerts with 67.2%), but also in particularly silent settings. In the latter case, also the notification LED is used (50.8%), as even vibration might be too disruptive in restaurants or libraries. Output modality usage also differs between tasks. The loudspeaker is particularly used for navigation applications (70.5%) and to notify on incoming calls (68.9%). However, for incoming calls, haptic feed-



	Screen	Loudspeakers	Headphones	Haptic Feedback	Notification LED
In public transport	80.3%	6.6%	55.7%	70.5%	42.6%
At home	82.0%	80.3%	34.4%	59.0%	52.5%
In a library/meeting	77.0%	0.0%	19.7%	45.9%	50.8%
During exercising	39.3%	34.4%	54.1%	41.0%	19.7%
In a restaurant	80.3%	3.3%	14.8%	54.1%	42.6%
Driving	37.7%	62.3%	21.3%	36.1%	31.1%
While walking	77.0%	27.9%	42.6%	60.7%	34.4%
On a concert	75.4%	8.2%	6.6%	67.2%	41.0%
Average	68.6%	27.9%	31.1%	54.3%	39.3%

(a) Situation-specific usage

	Screen	Loudspeakers	Headphones	Haptic Feedback	Notification LED
Notification about SMS or eMails	70.5%	42.6%	16.4%	68.9%	49.2%
Notification about incoming call	75.4%	68.9%	23.0%	72.1%	26.2%
Notification about due task or calendar	77.0%	37.7%	16.4%	62.3%	31.1%
Notification about system events	77.0%	24.6%	8.2%	49.2%	27.9%
Reading or being read a short text	73.8%	26.2%	26.2%	11.5%	6.6%
Reading or being read news	73.8%	19.7%	21.3%	8.2%	1.6%
While using a navigation application	73.8%	70.5%	21.3%	8.2%	4.9%
Average	74.5%	41.5%	19.0%	40.0%	21.1%

(b) Task-specific usage

Figure 6.2: Output modality usage in different situations and for different applications and use cases. The diagrams indicate percentages of agreement (white = low, yellow = medium, green = high agreement).

back is even more popular (72.1%). Vibration is likewise frequently used for other kinds of notifications, like emails (68.9%), to-dos (62.3%), or system events (49.2%).

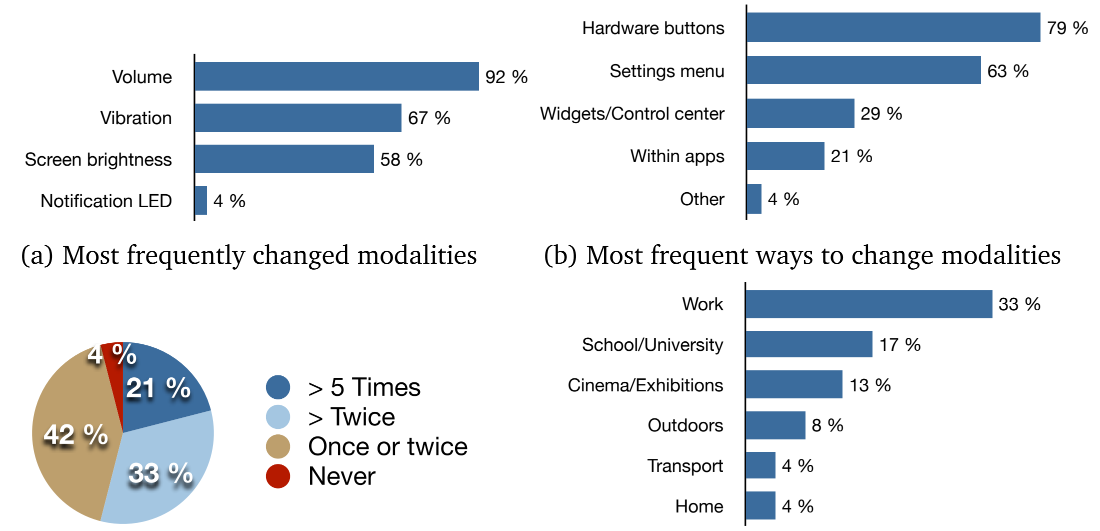
In summary, the results show that multiple modalities play a role for users when interacting with their mobile device. Modality usage is diverse, although the touchscreen is justifiably the most important channel for both in- and output. The question which modality is preferred is situation- and, especially for input, task-specific. This finding is a first motivation to facilitate switching between modalities on mobile devices. As we observed in particular for output modalities that their choice is remarkably influenced by different contexts and situations, this makes them particularly interesting for a further investigation to understand when and how automatic modality switches could be performed.

Change Behavior of Modalities

In a second survey, we investigated how, when, and how often subjects *change* modalities (with a focus on output modalities). We also were interested in subjects' attitudes towards automated multimodal behavior. 24 participants (6 females, 18 males) answered this questionnaire. They were aged from 20 to 31 years (M = 24, SD = 3)[80].

We first asked subjects which modalities they adjust most frequently, how often and in which situations these changes occur, and by which means they are performed. The answers to these questions are summarized in Figure 6.3.

[80] This survey was conducted as part of the user study described in Section 6.4.2.



(a) Most frequently changed modalities

(b) Most frequent ways to change modalities

(c) Frequency of modality changes per day

(d) Situations where modality changes occur

Figure 6.3: Current behavior of users with relation to multimodality. The diagrams indicate percentages of agreement. For questions (a), (b), and (d), multiple answers were allowed.

Among the often changed output modalities, the volume level was mentioned by far most frequently with 92%. This is comprehensible, since the aural modality is most obtrusive for others. Following were mentioned the activation/deactivation of vibration (67%), screen brightness adjustment (58%) and activation of the notification LED (4%). The fact that many participants in the study used iPhones (58%), which do not have dedicated notification LEDs (although the camera flash can be used for this function), might be responsible for the humble usage of this communication channel. See Figure 6.3a for a visualization of the results.

21% indicated to change modalities more than five times a day. 33% modify them more than two times a day. The majority of subjects (42%) stated to modify modality settings only one to two times a day, and 4% change them never (see Figure 6.3c). When changing modalities, the most frequently used method indicated by participants were hardware buttons (79%), e.g., the volume up/down buttons or the mute switch. This must be seen in relation to volume as the most frequently changed modality. 63% stated to use the settings/preferences menu; only 29% use shortcuts like the iOS control center, Android quick settings, or widgets. 21% change settings from within apps. One participant answered that he used NFC tags placed at different locations at home to trigger modality changes (see Figure 6.3b).

As most common situations in which inappropriate (i.e., socially not accepted) modalities were experienced, subjects mentioned work (33%), school/university (17%), and cinema or exhibitions (13%). Less frequently, subjects experienced modalities as unsuitable outdoors (8%), during transport (4%), and at home (4%). The latter case mostly happened when users forgot to unmute their phone when returning home after work (see Figure 6.3d).

We also asked whether subjects were satisfied how output modalities are prevalently handled on their mobile device. Did they rather miss information (i.e., notifications are too passive), or did they feel notified in an inappropriate manner (i.e., notifications are too obtrusive)? Overall, 58% of subjects were satisfied. Out of the remaining 42%, 25% found the modality too passive, 8% too obtrusive, and 8% inappropriate in both ways.

These results give first indications that a significant part of users is not fully satisfied with prevalent modality adjustment methods as used/available today. We argue that a solution as we present it in this thesis could improve this situation. Using rules for modality switches could be one solution to assist users who now frequently change modalities, and especially address the problem of forgetting to revoke a modality change. Although context classification has made significant advances, current machine learning approaches do not allow to autonomously manage context-aware behavior in the real world for a larger number of different situations with the necessary amount of precision [216]. Moreover, feeling in control is an important factor to user satisfaction [308]. Thus, manual rule definition seems a reasonable approach. This is further confirmed by our studies we describe later in this chapter.

Such a rule-based approach, however, requires that the smartphone gathers context information in the background and is able to autonomously change modalities. We therefore investigated subjects' attitude towards such a proactive behavior. We received mostly positive feedback on these questions (see Figure 6.4). Only 13% would not want their device automatically observing context factors. 67% would accept it without concerns, and 21% would accept it, but with some concerns, e.g., regarding increased battery consumption and affection of their privacy. Here, it was important to users that the gathered information would not leave the device. This is not the case for every context framework previously presented in literature, e.g., Code in the Air [273]. 64% would also accept without concerns that the device can automatically modify modality settings. 33% would accept it with some concerns, and only 4% do not desire this behavior. As concerns were mentioned here that the automatic switching between modalities might not work reliably enough, and that the user could lose control over the device.

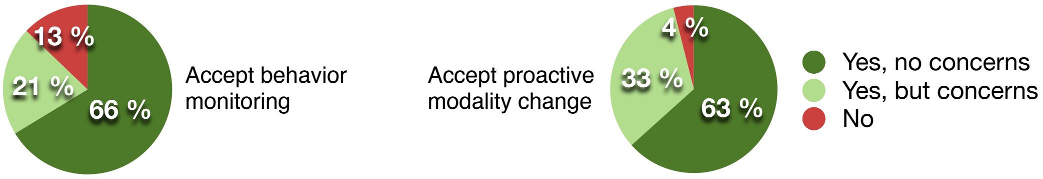


Figure 6.4: Acceptance of a system observing one's own behavior (left) and proactively changing output modalities (right)

6.2.2 Developer's Perspective: Expert Interviews

We interviewed three software developers[81] involved in mobile application development, asking how satisfied they are with the current tool support for creating multimodal applications. We also asked what they wished for to improve the support for their programming needs. Summarized and aggregated, the following issues were mentioned: Developers reported that implementing contextual behavior requires the use of different APIs (e.g., location API, sensor API, etc.). This does not only entail frequent reuse of similar pieces of code, but also heterogeneous ways of accessing data. As an example, sensor or location updates are listener-based

[81] Nielsen and Landauer [245] found that with small number (3–5) of target user group members, already the majority of problems of tested systems can be identified

("push" principle), while ambient light level must be checked manually ("pull" principle). If context-sensitive behavior based on push- and pull-based readings shall be realized, interfaces and wrappers need to be created, which adds significant overhead. Developers would appreciate a unified structure for all types of context information, as well as encapsulations of frequently used functions hiding complexity. In terms of input, novel interaction methods currently have to be designed and implemented from scratch. Especially for rapid prototyping, building blocks would speed up the creation of functional prototypes. In terms of output, it is effortful to include multiple modalities in an application, as each additional output modality must be implemented separately. If developers had a way to abstract from the information to be communicated and the channel over which it is transported, they would more likely implement multiple modalities, contributing to more usable, natural, intuitive, and efficient applications.

6.2.3 User's Perspective: Focus Group

The focus group was conducted with six participants (5 males, 1 female) between 24 and 30 years (M = 27, SD = 2). Four participants were research assistants; two were students. Four of them owned an Android smartphone and two had an iPhone. All participants rated their smartphone expertise as "high" (4) or "very high" (5) on a five-step Likert scale. At the beginning, participants were introduced to the topic and an overview of input and output modalities was given. Subsequently, the above research questions were investigated in a guided discussion. The focus group took about one hour and was audio-recorded. In the following, we summarize the most important results and design implications.

Results

Current Usage of Input and Output Modalities

While the touchscreen was the prevalent interaction method, the usage of further modalities broadly varied between participants. One participant used vibration and notification lights as primary output; another participant did not use vibration at all, but heavily relied on speech input (although only in private space). A third participant neither used sound nor vibration, but relied on screen notification, keeping the phone next to him on the desk. This indicates that one should account for diverse preferences and defaults. Second, the focus group revealed that modalities are rarely changed but that participants maintain "default settings" that are only altered in certain situations. For example, one subject only enabled sound when expecting an important call. Another subject had the phone in ringing mode as default, but muted the phone in silent environments (e.g., in the library). This shows that defaults are interpreted differently. It also turned out that when speaking of modalities, participants had mostly output in mind. As input methods, other than the touchscreen, participants indicated to use, e.g., hardware buttons to control the music application or to decline calls, and speech input to set a timer or perform a search query. Input modalities are however strongly task-specific. A framework should therefore support the implementation of novel modalities for individual purposes.

Identified Problems in Status Quo

The following problems and desires for improvements could be identified from the participants' feedback: Less conventional input methods either do not work reliably enough (speech), are not equally usable in different social contexts, or are not implemented consistently (e.g., tap gestures or gaze-based interaction, as found in some Samsung phones). In that case, participants desire them to be available and work in the same way in all applications. One subject stated that he often forgot to revoke a modality change, which could result in unwanted situations (e.g., the phone rings in a silent environment). This fear was confirmed by almost all participants and given as a reason for rare changes of modalities. They would welcome a system that automatically changes modalities, but retain control and be able to override the system behavior. Users miss a solution similar to "profiles" (as available on earlier Nokia phones, e.g., the Nokia 6110), so that a rule-based modality switching approach seemed attractive, especially using context information as a basis for decisions. Participants supposed that most situations could possibly be covered by a small number of rules. This would made complex automated approaches unnecessary. However, a prerequisite is that rule application works reliably.

User Interfaces

In the discussion of potential user interfaces and methods to set up and control multimodal behavior, several ideas emerged. They build upon the rule-based approach, as suggested in the previous section.

- The user manually creates rules in a dedicated interface by defining modality settings, which are activated by conditions (such as location, time, etc.). Participants supposed that rules would in many cases be set once and later rarely changed.
- The user makes the desired settings in the phone and assigns this "profile snapshot" to a certain condition. The challenge here is to define which settings should be included to the rule, and which should not.
- The smartphone recognizes the interdependence between a certain situation (context) and associated modalities, learning from the user's actions, and proactively suggests new rules to automate modality switches.

Users always need to have the possibility to override these rules, with changes remaining active until the next rule is applied. Another idea was a time limitation for manual settings (e.g., one hour or *"until the meeting is finished"*). Users should also be quickly able to glance which setting is active. One possibility to achieve this would be a widget on the lock screen or home screen that informs the user on current modalities and active rules, right after switching on the phone. However, if the widget is not placed on the primary home screen, it would consume additional time to switch home screens. A notification message could be the better solution in this case. We further investigate these alternatives in a study in Section 6.4.2.

Based on focus group and expert interviews, we can summarize the following requirements:

1. A holistic system for mobile multimodal interaction should both cover multimodal input (in terms of natural interaction) and output (in terms of context-based modality switches).

2. Defining multimodal behavior based on rules is closest to the human understanding of automated switching and might thus be an adequate underlying model, in contrast to machine learning.
3. The system must be flexible enough for the heterogeneous preferences in terms of favorite modalities in different situations and for different applications.
4. The underlying programming framework should ease the access to contextual information in a unified way, and abstract from information representations (i.e., a unit of in-formation can be easily communicated by different modalities).

6.3 Software Framework for Multimodal Interaction

In order to address the requirements formulated above, we implemented a framework to support the development of multimodal interaction. Our M3I (Mobile MultiModal Interaction) framework is implemented in Java as Android library. Thereby, no special configuration or tools are needed for using it; the library just needs to be referenced from an Android project to access its features. We share M3I with the community at `http://www.eislab.net/m3i`. Figure 6.5 shows, on a conceptual level, the structure of the framework and its basic components, which will be detailed in the following.

6.3.1 Rule-Based Modality Switches

M3I defines the multimodal behavior of a system with the help of rules. We opted for the rule-based approach out of the variety of possible paradigms discussed in Section 2.3.2, based on the preceding focus group discussion (see Section 6.2.3). We acknowledge that static rules may provide limited flexibility compared to, e.g., self-learning approaches. However, in the focus group we found that subjects intuitively suggested (simple) rules to determine autonomous behavior. We therefore argue that this approach might comply best with the underlying mental model and the understanding humans have of autonomous systems. Further, user-defined rules support the feeling of control and the understandability of modality changes. Some possible alternatives to manual rule creation will be discussed in Section 6.4.1.

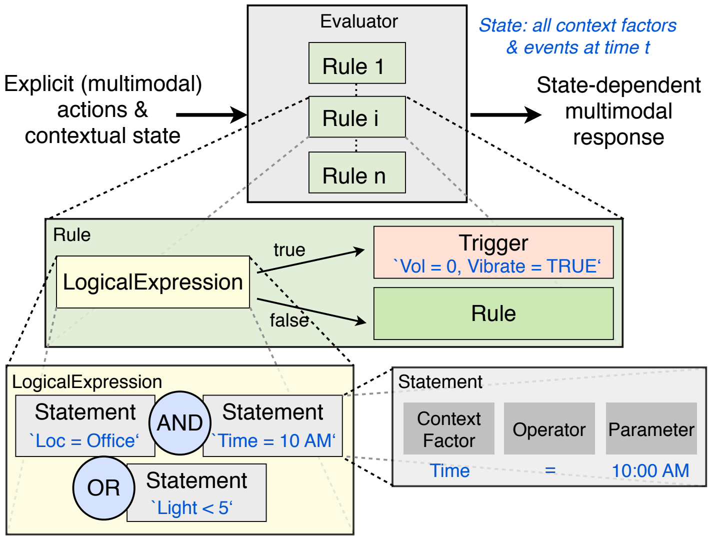
As shown in Figure 6.5, rules have a “if-then-else” structure. This simple approach is motivated by making them adequate for non-expert programming[82]. In each rule, logical expressions, which define a certain (context) situation, are evaluated to be either true or false. In either case, a defined action can be triggered, or another rule can recursively be called. This allows creating a nested decision tree of arbitrary complexity. The set of active rules is evaluated in the framework’s `Evaluator`.

For each rule, evaluation intervals can be set, in order to account for time-critical sensor data as well as for battery-conserving location updates. Once rules have been defined, the follow-

[82] cf., e.g., programming systems addressing beginners or children, such as Lego Mindstorms, `http://mindstorms.lego.com`, accessed September 11, 2014

ing code excerpt activates the framework in any Android application. This demonstrates how easy it is to extend existing applications by the functionality of M3I.

```
Evaluator e = new Evaluator(1000);  // Create new evaluator instance
  e.addRule(rule_1);                // Add previously defined rules
  ...
  e.addRule(rule_n);
  e.start();                        // Check in the background if rules apply
```



Example: If I am in a meeting (at the office AND at 10 AM)*, OR if the phone lies with the screen facing down* (here tested by 'ambient light level < 5' as the light sensor is on the front side)*, mute the phone and enable vibration.*

Figure 6.5: General Structure of the M3I framework. The core is the evaluator, which evaluates rules that define the system's behavior in response to explicit user actions and/or to implicit context factors and events. Rules can initiate triggers to configure modality changes for in- and output and to adapt user interaction to the context. Recursive rules allow the implementation of complex decision-making. Examples are shown in blue, smaller font.

6.3.2 Context Integration

So-called context factors are fine-grained pieces of context information that can impact output modalities, or are part of an interaction (e.g., a gesture). Examples for context factors are the device's location, the charging state of the battery, or the time of day, but also complex information such as the user's current mode of transportation or activity. In M3I, a `ContextFactor` is defined by specifying a `ContextGroup` it is part of, and the respective context method that provides the `ContextFactor`'s value. Since Java does not support function pointers, constants are used to specify the ID of the context method that shall be called at runtime. For example, a context factor of type `Float` indicating the device's charging level can be defined as follows:

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

Methods for retrieving context information are organized in groups such as OrientationContext or LightContext, which on their part contain information on the device's orientation or the ambient light level. Context groups can be seen as collections that organize and provide access to the platform's own methods and routines in a way that eases building up statements and rules. The framework drastically economizes and simplifies those method calls for the programmer, compared to using system APIs directly, as it saves all overheads for initializing system services, for creating event listeners etc., and makes all functionality accessible by homogenous interfaces. One particular advantage of M3I is the unified handling of synchronous and asynchronous information. Values available anytime (e.g., the current weekday) can be retrieved with a simple method call, but dynamic events are normally handled by listeners asynchronously when they occur (e.g., location changes, sensor events, or touch interactions by the user). The framework makes transparent to the developer whether a ContextFactor is based on a synchronous or an asynchronous call (pull vs. push). It internally creates listeners if required, handles their updates automatically and stores the most recent values. Thus, whenever the contextual state should be determined, a consistent state is available. Events can be set to expire after a defined period of time, or to have unlimited validity. By combination of validity and event timestamps, occurrences of sequential actions (e.g., for multi-step motion gestures) can be evaluated.

6.3.3 Decision Logic

Triggers define what should happen for a defined contextual state. They realize modality switches, i.e., they abstract how information shall be represented. For example, one and the same notification can be provided as sound, textual message, or vibration. A number of modalities are directly supported, e.g. visual (UI changes), haptic (vibration patterns), or auditory (sound playback) responses. Triggers are not limited to predefined modalities or actions: A MethodTrigger calls an arbitrary method to implement custom functionality. With the NullTrigger, the omission of an action, e.g., in an else branch, can be modeled. As an example, a trigger that enables vibration mode can be defined as follows:

```
VibrationTrigger vibrate = new AudioTrigger(this);
vibrate.setAction(AudioTrigger.RINGER_VIBRATE);
```

System-Wide State Announcement

StateTriggers are special triggers that announce states or contexts system-wide through an Android content provider. That way, each application on the phone can decide how to adapt to state changes, accounting for the fact that interaction modes are often task-specific. For example, in a mobile context (e.g., when the user is walking), an application can offer a UI with larger touch controls that are easier to hit.

Logical Expressions

The simplest form of a logical expression is a Statement, which consists of a ContextFactor and an Operator. Operators allow numeric comparisons, but also within-range-tests, or regular expressions. The following code example shows a “greater-than” operator, which checks whether the device is charged more than 50%. The statement uses the context factor battLevel that has been defined in a previous example.

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

Using logical operators realized by UnaryExpression and BinaryExpression classes, complex terms can be created. Although AND, OR, and NOT are sufficient to construct any logical expression according to Boolean algebra laws, XOR, NAND, NOR and XNOR are supported for advanced users to implement a decision logic of arbitrary complexity. In the following code example, we describe the state in which the device is either charged more than 50% or plugged in (in the example, it is assumed that isAboveHalfCharged and isPluggedIn are previously defined statements).

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

When triggers have been defined and the contextual states in which they should be applied have been described by logical expressions, a rule can be created. For a code example, see the first described application in Section 6.3.4.

Extensibility

M3I’s flexible input and output wiring mechanism goes beyond context-sensitive programming, and opens up a huge design space for multimodal interaction methods, as, e.g., gestures can be linked with arbitrary actions (see Section 6.3.4 for some examples). With a CustomContext group, results of any method or callback can be fed into the framework’s decision logic when a context factor is not built in directly. By clearly defined interfaces, the framework is easy to extend both on the input and output side. The following simple steps are, e.g., required to add a new context group:

- Create a new class implementing the interface IContextGroup, and implement all context methods the group should provide, returning a value of arbitrary data type.
- Provide method IDs for each context method so that they can be called by the execute() function of the group. In a similar manner, new triggers implement the interface ITrigger, which contains a trigger() method that performs the desired action.

M3I currently integrates more than 50 context factors regarding, e.g., location, ambient noise and light level, device orientation, battery information, proximity information (through NFC, Bluetooth, or Geofence entering/leaving in conjunction with the within operator), availability of 3G and WLAN connections, or date and time. Unified access to several Android APIs, basic activity recognition and classification routines abstracting from pure sensor readings are already integrated, e.g., pose classification (in pocket or carried in hand), usage indicators, mode of transportation, vision-based detection (face), etc. Besides that, explicit interactions, such as physical button presses or touch interactions can be intercepted and combined with implicit contextual information. On output side, triggers allow controlling a range of modalities and thereby abstracting how information should be represented, based on fine-grained definitions. Examples include visual output (on-screen or via LEDs), sound, screen brightness setting, or vibration. Further actions include behavioral triggers, e.g., changing device settings (sync rules, connectivity, screen lock, etc.) or custom functions.

Thus, given that respective sensors are provided by the device (see Section 2.1.2), M3I supports "classic" modalities corresponding to human senses (e.g., vision, audition, touch, thermoception), as well as "combined" modalities in terms of novel interaction methods. Some examples for this will be presented in the subsequent section.

6.3.4 Exemplary Validation of the Framework

To validate our claim that M3I fosters and simplifies the development of context-based multimodal applications and interaction methods, we present two different applications that give an idea of the spectrum of potential use cases for our framework.

Physical Interaction

This basic application uses a "mute by flip over" device gesture for quickly enabling silent mode, e.g., in a meeting. It is thereby an example for physical or tangible interaction [81, 85]: Instead of manually turning down the volume, the user simply needs to flip over the phone and place it on the table with the display facing down. A dozen years ago, such context-based telephony applications were prototyped with external sensor modules [297]. Even today, with common means, this demo would require considerable coding effort of several dozens of lines of code (listen for sensor readings in the background, implement modality change from scratch, etc.). With M3I, the implementation is quite simple: We realize the trigger to mute the phone by checking the light level. The ambient light sensor lies on the front panel of the phone, so that we assume that the phone is turned upside down if the sensor reading falls beyond a certain threshold.

```
// define statement
LightContext lc = new LightContext();
FloatContextFactor light = new FloatContextFactor(
  lc, LightContext.FLOAT_GET_LIGHT_LEVEL);
Statement isUpsideDown = newStatement(light, FloatOperator.smallerThan(5.0f));
// define triggers
AudioTrigger mute = new AudioTrigger(this);
mute.setAction(AudioTrigger.RINGER_VIBRATE);
AudioTrigger ring = new AudioTrigger(this);
ring.setAction(AudioTrigger.RINGER_NORMAL);
// finally, put rule together
Rule r = new Rule(isUpsideDown, mute, ring);
```

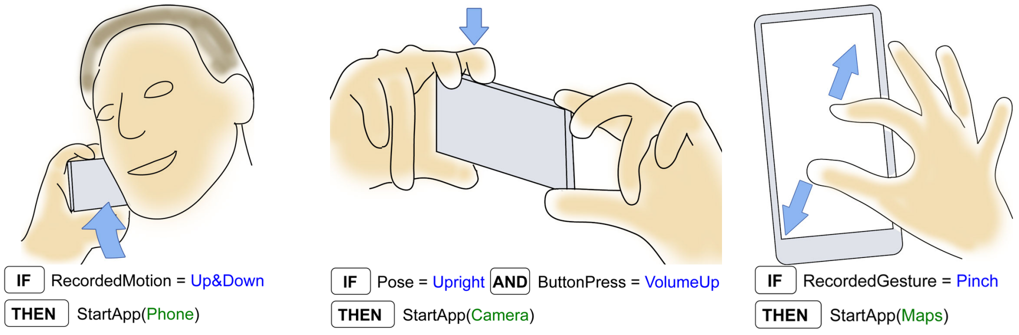
As this example is deliberately kept simple, we do not take all possible cases into account. We cannot distinguish if the user has turned the device with the screen facing down, put it into a sleeve, or if it is just dark in the room. However, the rule could easily be refined by adding more context factors, e.g., the time of day and the pose. Nevertheless, the example demonstrates how few lines of code are sufficient to realize a multimodal input method.

Mimikry Input

As multimodal input is (as argued earlier), in contrast to output, task-specific, we implemented three unique methods (*Raise to Call*, *Press to Shoot*, and *Pinch@Home*) to launch different applications (Phone, Camera, Maps). As our goal was high intuitiveness, we chose metaphors that support the human mental model of performing typical movements with these apps. Each method includes different modalities, like motion gestures (performing a characteristic motion with the device), screen gestures (performing a multi-touch gesture on the screen), or explicit voice or button input. In the following, we describe the interaction methods in more detail:

- **Raise to Call** (Figure 6.6a): This interaction method launches the phone app. The user raises the phone to his ear and speaks the name of the person to call. The mimicked movement is the gesture when answering a call. The method involves the *motion gesture* modality, which was recorded using M3I.
- **Press to Shoot** (Figure 6.6b): This interaction method launches the camera app. The user brings the phone to her front in an upright position and presses the volume button, mimicking the typical movement when taking a photo. This method includes two parallel input modalities: a motion gesture and a button press. Both modalities were combined in a logical expression using the AND operator.
- **Pinch@Home** (Figure 6.6c): This interaction method launches the maps app. The user uses thumb and index finger to perform a multi-touch pinch gesture on the home screen or lock screen, as if zooming into a map. Here, the touch interaction modality is involved.

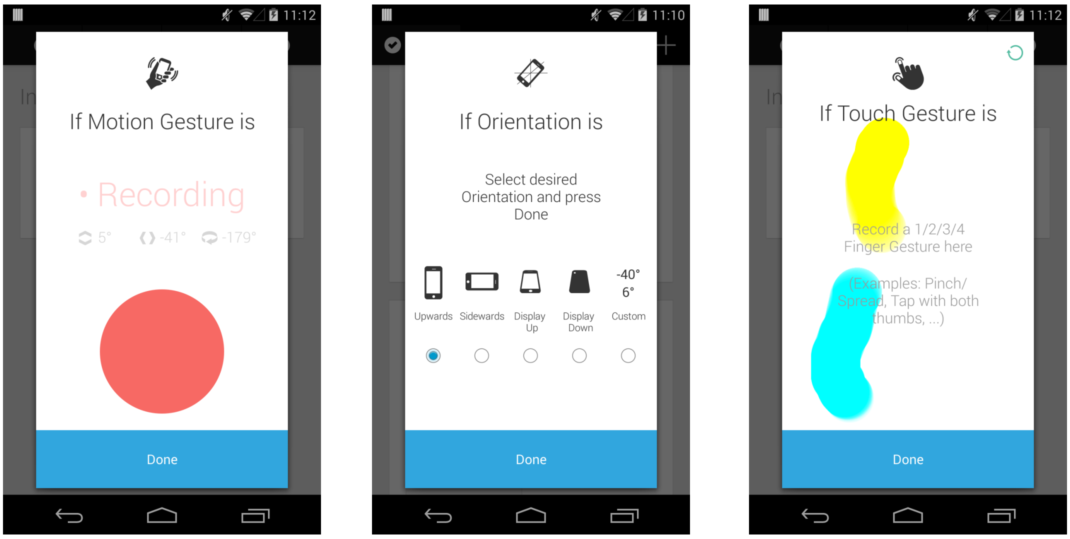
Figure 6.6 illustrates the input methods and additionally shows the rules by which they have been defined in the M3I framework.


IF RecordedMotion = Up&Down
THEN StartApp(Phone)
IF Pose = Upright AND ButtonPress = VolumeUp
THEN StartApp(Camera)
IF RecordedGesture = Pinch
THEN StartApp(Maps)

(a) Raise to Call (b) Press to Shoot (c) Pinch@Home

Figure 6.6: Mimikry gestures for launching applications implemented with the M3I framework

For each rule, the different input events were associated with a `StartApp` trigger to launch the respective application. All input methods were realized using the graphical user interface (GUI) which we will introduce in Section 6.4. This GUI simplifies the definition of motion gestures (Figure 6.7a) or multi-touch gestures (Figure 6.7c), and built-in abstractions (such as predefined device poses as shown in Figure 6.7b) additionally accelerated the development of the *Raise to Call* and *Press to Shoot* methods.



Select desired Orientation and press Done

Record a 1/2/3/4 Finger Gesture here

(Examples: Pinch/ Spread, Tap with both thumbs, ...)

(a) Recording a motion gesture

(b) Defining the device orientation

(c) Recording a multi-touch gesture

Figure 6.7: Defining multimodal behavior through the rule creation user interface on top of M3I

Multimodal Game Controller

This example shows how to use a mobile phone as game controller for a car racing game running on a second screen. The vehicle can be steered by tilting the phone, as visualized in Figure 6.8a). The angle of the steering wheel is derived from the *pose detection* context factor, which can directly retrieve the tilt angle of the device using the accelerometer or gyroscope.

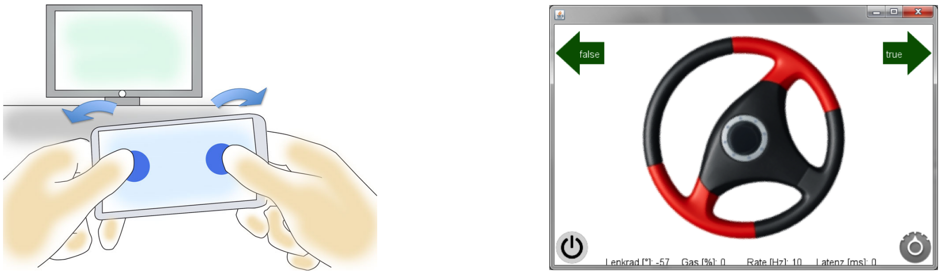
In addition, soft buttons on the screen can control further functions, like accelerating and braking. The game controller scenario combines the input modalities *touch* and *device gestures*. It further demonstrates the framework's capability to seamlessly integrate explicit (button presses) and implicit events (updates from the orientation sensor listener). The communication to the game running on an Ubuntu desktop PC is realized using ROSjava[83], a publish/subscribe architecture based on the Robot Operating System (ROS) [267], a popular architecture for distributed applications in research (see, e.g., [48, 85, 178]). Figure 6.8b shows a screenshot of a corresponding client application on the PC that visualizes the inputs made by the remote control application. As an example for a practical use case, the presented setup can control an open-source driving simulator such as OpenDS[84].

Tasking Application

While developers can use M3I for a variety of individual use cases, it is also desirable to make the full potential of M3I available to end users. In Section 2.3.2, we have given examples for tasking applications that automate the modality of, e.g., system notifications according to the context. While such tasking applications (available, e.g., in Google's Play Store, see Section

[83] `http://code.google.com/p/rosjava/`, accessed March 5, 2014

[84] Open Driving Simulator, `http://www.opends.eu`, last accessed April 3, 2014



(a) Remote control setup: tilting and action buttons steer a car in a PC application

(b) Controlled client on remote PC exemplified by a steering wheel

Figure 6.8: Implementation of a multimodal game controller with M3I

2.3.2) have been programmed from scratch, one could easily implement similar and even beyond-going functionality with M3I. We therefore created a GUI on top of the logical structure of the framework, allowing end users to create rules based on the previously presented concepts (statements, logical expressions, triggers, etc.) in a straightforward manner. Figure 6.9 shows the basic concept for rule creation, where conditions (i.e., statements and logical expressions) are set on the left side, and triggers on the right side. The image exemplarily shows the creation of the *Pinch@Home* input method.

In concordance with a suggestion made by the focus group, the system goes back to the previous setting when a rule does not apply any more. That way, individual default settings are possible when no rule is active, which accounts for heterogeneous preferences of users. For this proof-of-concept implementation, we did not include a check for conflicting rules. However, there exist approaches for automatic verification of rule systems [335].

A further design question was the realization of nested decision logic in the GUI, i.e., an intuitive way for entering logical expressions with multiple nested AND/OR operators. We wanted to help users keep track of hierarchical levels while sparing them the effort of entering opening and closing brackets for entering an expression like *"(A and B) or C"*. To this end, the interface offers two sets of AND/OR buttons (see Figure 6.9). While the blue button set adds a new statement in the inner level, the white button set adds a statement at the outer level. For example, adding statements with the blue AND button and the white OR button, the expression *"(Two-Finger Pinch and B) or C"* would be created.

Figure 6.7 shows some dialogs of this tasking application that assist the user in creating input modalities including, e.g., motion gestures, screen gestures, or device poses. The main user interface of the tasking application reflects the rule-based approach of defining multimodal behavior. In Section 6.4, we will deeper investigate different user interface options to accomplish rule creation, and evaluate them in a user study.

Using M3I for Implementing Research Apps

To further illustrate the applicability of M3I in a research context, in the following we outline opportunities of utilization of M3I in the development of systems we have presented in Chapters 3, 4, and 5.

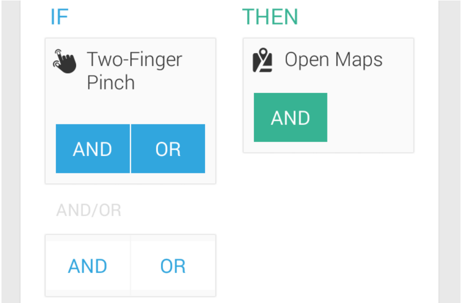


Figure 6.9: Graphical user interface (GUI) to assemble a rule. The left part contains the logical expressions representing the contextual state. In the right part, the trigger for the respective action(s) is defined.

Medication Package Recognition (see Chapter 3.2)

The MobiMed application used different modalities (touching, pointing, scanning) to recognize drug packages. The basic mechanism of coupling an input method with the according output (i.e., the description page of the medication) could likewise have been realized by M3I. However, M3I would have to be enhanced by the possibility to add parameters to triggers (the medication ID), so that the correct information can be displayed.

As a possible extension of the MobiMed scenario, the application could launch automatically when the phone is held in a special way (e.g., raised and targeted at a drug package). This usability enhancement supports the target group of elderly people, who might have difficulties locating the application in the launcher menu or touching small application icons. A challenge is that such a *motion gesture* modality could theoretically overlap with other actions on the phone. However, given that elderly people often use only a limited number of applications, it would be possible to design the modality in a way that it does not interfere with the (known) set of interaction methods used otherwise, and is not accidentally triggered. With M3I, the motion gesture could be realized using the pose detection in combination with a location context, so that the trigger is only activated at home, but, e.g., not when traveling.

Physical Exercise Assessment (see Chapter 3.3)

In our exercise assessment use case, we had incorporated an automated recognition of the training device using NFC when the smartphone was placed on the rocker board. That functionality could be taken over by M3I, using an application launch trigger upon recognition of a valid NFC tag. The GymSkill application incorporated two types of feedback: simple responses during training, and a detailed assessment after the performance. A simple pose classification could possibly be realized with M3I (such as alarming the user on maximal deflection angles of the rocker board). Visual and auditive notifications would simply be triggered by pre-defined poses. However, the more complex exercise assessment algorithms go beyond M3Is capabilities and must still be implemented separately.

University and Education (see Chapter 4)

In the concept of our didactics method toolbox MobiDics, context plays an important role. The context-dependent selection of appropriate teaching methods could be simplified with

M3I's context abstraction, e.g., to classify different locations on top of the underlying location provider. The recognized context classes could then be fed to the application and used to inform the method selection algorithm (e.g., confining the search to methods appropriate for the respective course type). Alternatively, this classification could be made based on time instead of location. The actual course taught could be retrieved from the user's timetable, and MobiDics could be adapted respectively. Rules based on weekday and the time of day are likewise directly supported by M3I. Furthermore, if no room-level localization was used (requiring an accurate indoor location provider), a more coarse location distinction (e.g., *"at the university or at home"*) could be used for different presentation modes. As an example, in classroom mode, the user interface could be confined to method aspects that are important for "live use" and hide additional details to remove distraction. Further, the device could be automatically muted if the docent looks at multimedial method instructions in class which contain sound. By contrast, in preparation mode at home or at the office, all methods and instructions are available in full level of detail for deeper studying.

Indoor Navigation (see Chapter 5)

Various aspects of the indoor navigation prototype could be realized with M3I. First, the pose-dependent switching between AR and VR mode, as described in Section 5.2.2, could be managed by a rule in M3I. Further, the framework could define swipe gestures to switch between VR and map view. Touchscreen gestures could also enable further settings like a mode for context-aware services, in which users can interact with points of interest.

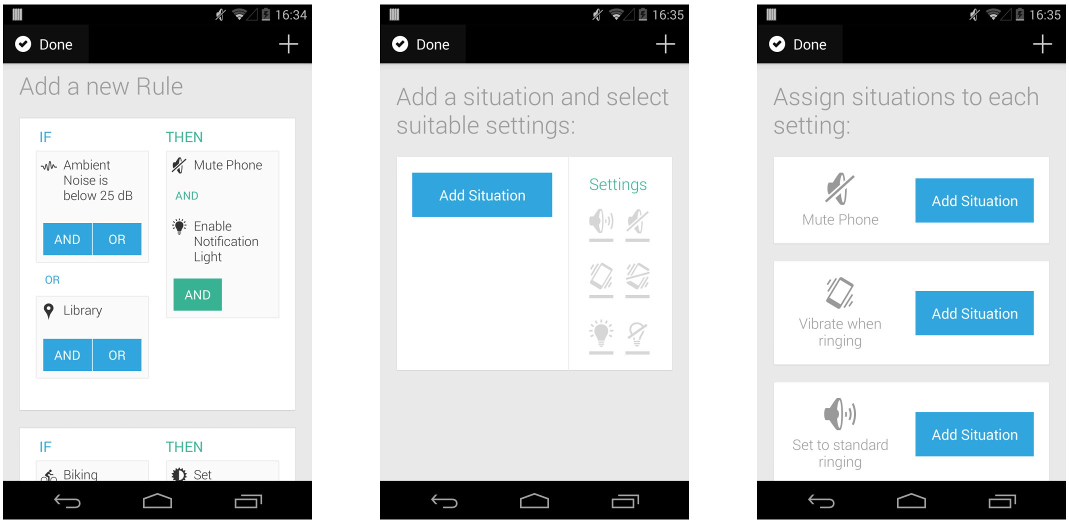
Besides that, context could be used to influence the output modalities. The required context factors for this scenario are supported by M3I. First, walking speed could be used to adapt size and level of detail of navigation instructions. When the user walks faster, instructions are enlarged and contain less details. When the user slows down, the level of detail is increased, as the user is potentially uncertain or is interested in exploring nearby POIs. The noise level could serve as an indicator whether the application should use speech or visual output to guide the user. Ambient noise could either be detected using the microphone, or pre-defined geo-coded settings could be used.

6.4 User Interfaces to Define Multimodal Behavior

In this section, we present and evaluate different approaches and user interfaces for defining multimodal behavior as supported by the M3I framework (see Section 6.3 for a description of the framework). We first introduce different user interface concepts and subsequently report on the laboratory and field study we conducted.

6.4.1 Concepts

We suggest three different variants to define multimodal behavior, whose concepts have partly been informed by a preceding focus group discussion (see Section 6.2.3).



(a) Balanced interface (*BAL*)

(b) Situation-oriented interface (*SIT*)

(c) Modality-oriented interface (*MOD*)

Figure 6.10: Three user interfaces to *manually* define rule-based multimodal behavior

Manual Rule Creation

The first approach is the manual creation of rules, which reflects the functionality of the decision logic provided by the underlying toolkit in the most direct way. In Section 6.3.4, we have presented an initial implementation and UI of this concept. Here, we refine this approach and discuss three different UI variants: a balanced, situation-oriented and modality-oriented interface (see Figure 6.10).

Balanced In the balanced interface (see Figure 6.10a), the determining (context) factors and triggers are configured in an equivalent manner. In the description of the end user toolkit in Section 6.3.4, we have already briefly outlined how to create nested expressions with AND and OR operators. For reasons of simplicity, we confined the user interface to a maximal nesting depth of one. That means it is, e.g., possible to create an expression *"if (A and B) or (C and D)"*, but not *"((if A and B) or C) and D"*. As we will show in the following user study, subjects did overall not miss the possibility to create more complex rules, so that the present possibilities cover the majority of cases needed in practice.

As further simplification, we omit the unary NOT operator when combining statements to logical expressions. Instead, the NOT operator can be selected together with context factors. For example, if a `Time` context factor is added, the user can choose to add the statement *"within time interval i"* or *"not within time interval i"* to the expression.

Situation-Oriented The situation-oriented interface (see Figure 6.10b) puts the focus on situations or contexts, to which the user assigns different modalities. This makes rule creation especially easy if multiple modalities shall be changed together in a certain situation (e.g., when arriving at a certain location). This interface is, however, less suited when identical modality changes shall be applied in different situations. Especially if these situations are, in turn, defined by complex logical expressions, it may occur that the limited nesting depth (as discussed above) does not allow to combine multiple situations with an OR operator. In that case, it can happen that the user has to create separate rules, and that the desired trigger must be defined multiple times (once for each rule).

Modality-Oriented The modality-oriented interface (see Figure 6.10c) is the counterpart of the situation-oriented approach. It supports modality-oriented thinking, e.g., *"in which situations do I want to mute the phone?"*. It contains a list of available output modalities, for each of which according contexts can be defined. This makes it easy to group similar situations (both mentally and in the interface) in which the multimodal behavior shall be the same. As drawback of this solution, contexts must be defined multiple times if more than one modality shall be changed at a time.

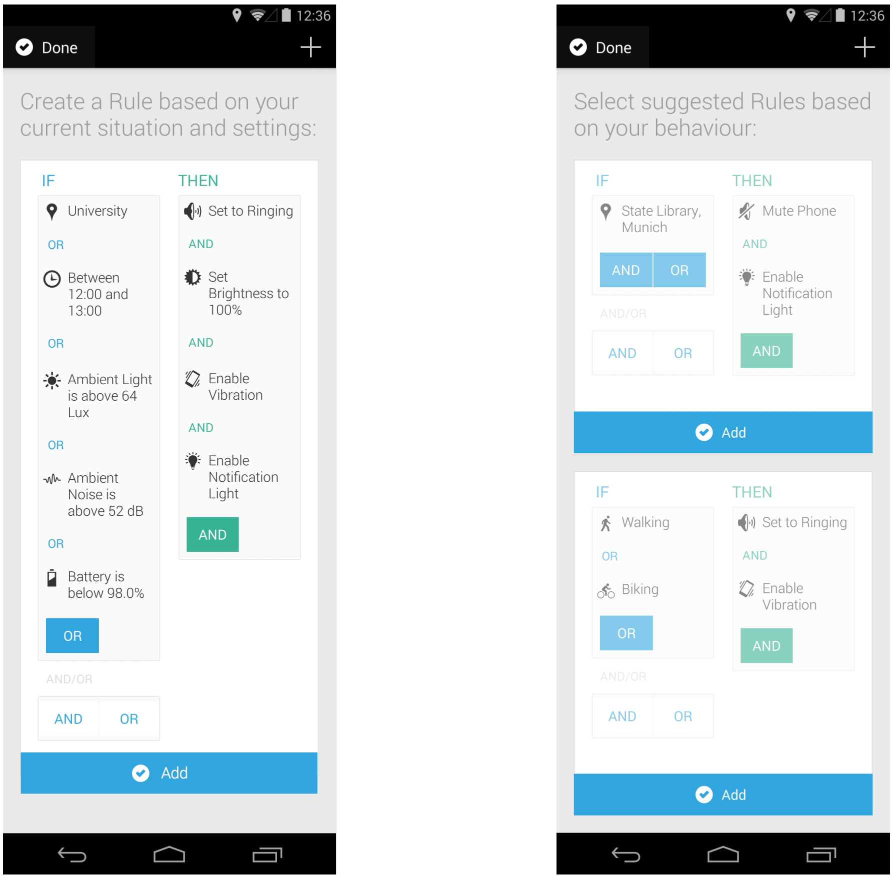
Summing up, the situation-oriented and modality-oriented interfaces provide a certain degree of simplification compared to the balanced interface, as no logical expressions need to be created from scratch on trigger side. However, they may be disadvantageous in certain situations. The balanced interface provides the most flexibility, but is also more complex to handle. We will evaluate efficiency and effectiveness of all three approaches as well as user preferences in a user study in Section 6.4.2. In that section, we also present example rules for which either the situation- or the modality-oriented approach is advantageous.

Snapshot

The *Snapshot* approach is designed to simplify the above process, especially in terms of *familiarity*, one of Shneiderman's eight Golden Rules of interface design [308]. It is motivated by the idea that a familiar UI should be reused as often as possible. Therefore, users do not have to define the context settings and appropriate modalities within the possibly unfamiliar UI of the rule creation application. Instead, they can make their settings as they know it, using the possibilities provided by the operating system (e.g., hardware buttons, the settings menu, widgets, or notification bar shortcuts). When the user wants to create a new rule, a snapshot of the beforehand made settings is taken as a starting point (see Figure 6.11a). For example, if the user is at the library and has previously muted the device, the rule *"Mute my device when at this location"* will be suggested. A challenge of this approach is, however, that potentially not all information included in the snapshot might actually be desired to be added to the rule. Hence, the user will have to select a subset of the suggested "candidate" context factors. In the prior example, the system might have suggested the rule *"Mute my device when at this location at 9:30 AM"*, although the user desires that the rule applies also at other times. Further, not all possible context factors can be determined automatically by the *Snapshot* method, so that some extent of manual editing will always be required. However, compared to creating a new rule from scratch, the necessary time for setting up rules can possibly be reduced.

Suggestion

The *Suggestion* approach analyzes the user's behavior and deduces rules from her actions. This can be realized by classification using machine learning techniques [161]. The main difference to the *Snapshot* approach is that here suggestions are made proactively. The user



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%

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

Select suggested Rules based on your behaviour:

IF
State Library, Munich

THEN
Mute Phone
AND
Enable Notification Light

IF
Walking
OR
Biking

THEN
Set to Ringing
AND
Enable Vibration

(a) The *Snapshot* approach offers current settings and contextual states that could potentially be part of a rule.

(b) The *Suggestion* approach proactively proposes rules based on a user behavior analysis.

Figure 6.11: Semi-automated approaches to accelerate and simplify the creation of modality adjustment rules.

receives for example a notification like *"You have several times muted your phone at the library. Shall an according rule be created?"*. Figure 6.11b illustrates examples for suggested rules. The advantage of this approach is that ideally very little user action is required. However, it bears the risk that suggestions are inaccurate and require a high amount of editing, making them almost as effortful as creating rules from scratch. In addition, users can be enervated by the proactive behavior if they do not wish to create a new rule at all at this time. In the focus group discussion (see Section 6.2.3), we found that most users might get along with a small number of rules. In light of this finding, suggestions must be made in a conservative way, so that the negative effects do not overweigh.

6.4.2 Laboratory Study: Effectiveness, Efficiency and Usability

We conducted a laboratory study to quantitatively evaluate the above described concepts with relation to effectiveness, efficiency and usability. Furthermore, our goal was to collect qualitative user feedback.

Research Questions

We investigated the following research questions:

RQ1 Which concept is most efficient?

RQ2 Which concept is most effective?

RQ3 Which concept is best in terms of usability?

These first three research questions address the manual rule creation approaches. We compared the balanced, situation-oriented and modality-oriented approach (in the following abbreviated as *BAL*, *SIT*, and *MOD*). Efficiency corresponds to task completion time, while effectiveness comprises the success rate (how many of the in total created rules were correct), and the error rate. For RQ3, we measured ease of use and clarity, and we used the SUS score as a measure for overall usability.

RQ4 What is the user acceptance for semi-automated rule creation approaches?

For this research question, we demonstrated the *Snapshot* and *Suggestion* approach to users and asked them whether they would like to use these systems in practice.

RQ5 What is user feedback on the system?

We used open-ended questions to get insights on subjects' opinions on the rule-based approach, and asked them to name rules they would create for their personal everyday needs (both for input and output). Furthermore, we asked for general acceptance and their attitude towards the approach.

Method

Task

Each participant evaluated the *MOD*, *SIT*, and *BAL* approach (within-subjects design with three conditions). In each condition, two different rules had to be created, resulting in six rule creation tasks for each participant. The order of conditions was counterbalanced using a Latin square design; the order of rules to be created was alternated.

For the tasks, the following two rules had to be created:

- **Rule 1**: When I am at the university, mute the phone and enable vibration.
- **Rule 2**: When I am biking or in a loud environment, set the notification to "ringing".

For Rule 1, the location "university" was predefined, so that users did not have to locate the university building on the map, which would have been a confounding factor. For Rule 2, we added in the description that "loud environment" means a noise level of 100 dB or more. Rule 1 is an example for two triggers (combined with an AND expression), while Rule 2 is an example for two context factors (combined with an OR expression).

Table 6.1 illustrates the minimal number of steps required to create Rules 1 and 2. While the modality-oriented interface has a slight advantage when different situations are assigned to one modality (as in Rule 2), the situation-oriented interface is slightly advantageous when several modalities are assigned to one situation (as in Rule 1). That way, neither interface should be significantly disadvantaged a priori.

Rule	Rule 1: *When I am at the university, mute the phone and enable vibration.*	Rule 2: *Set sound to ringing if I am on the bike or it is very loud.*
Structure	Situation A ↔ Modality X and Y	Modality X ↔ Situation A or B
Optimal Taps	*SIT*: 6 *MOD*: 8 *BAL*: 10	*SIT*: 8 *MOD*: 7 *BAL*: 10

Table 6.1: Rules used for the rule creation task of the laboratory study. The row "Optimal Taps" shows the minimum number of required taps to create the rule in either condition.

Measurements

For RQ1–3, data was collected through logging, analysis of a video recording of the participants performing the task, and questionnaires. We measured efficiency (task completion time), effectiveness (errors and success rate), and satisfaction (ease of use, clarity, and SUS [34]). The measurement of task completion time was triggered manually by the participant through pressing a button. The measurement was stopped when the participant pressed "Done" after the rule creation process. During rule creation, the task description remained visible in the upper part of the screen. Therefore, we had extended the interfaces visualized in Figure 6.10. Each deviation from the ideal path in the rule creation process was considered an error. A task was counted as successful if the created rule after pressing "Done" was correct (i.e., if no errors were made, or the errors were corrected). Errors and success rate were determined by video analysis. After each task, participants rated how easy it was to create the particular rule with the respective interface. Ratings were made on a 5-point Likert scale. Further, subjects answered a SUS questionnaire after each condition. User feedback for RQ4 and RQ5 was likewise collected by questionnaires which subjects filled out on a laptop in the laboratory.

Participants

24 participants (mostly students) took part in the study (6 females, 18 males). They were aged from 20 to 31 years (M = 24, SD = 3). All of them were smartphone owners. 14 owned an iPhone using iOS, 8 used Android-based phones, and 2 had Windows phones. Subjects' self-estimated smartphone expertise ranged from average to expert. On a 5-point Likert scale (1 = no idea of technology, 5 = expert), the average rating was 4, with each one third of participants choosing 3, 4, and 5.

Results and Discussion

RQ1: Efficiency

With a two-way repeated-measures ANOVA, we found a significant effect of approaches (F(2, 46) = 8.08, $p < 0.001$, partial $\eta^2 = 0.260$) but not of rule type (F(1, 23) = 4.09, p = 0.06, partial $\eta^2 = 0.151$) on task completion time. *SIT* was the fastest method with averagely 17.9 s for Rule 1 and 26.5 s for Rule 2. With *MOD*, subjects needed in average 29.8 s (Rule 1) and 29.3 s (Rule 2); with *BAL*, mean values were 28.7 s (Rule 1) and 34.8 s (Rule 2). The results are visualized in Figure 6.12. Post-hoc t-tests with Bonferroni correction revealed significant differences between *SIT* and the other conditions, but not between *MOD* and *BAL*.

Analyses of the video recordings showed that with *SIT*, users had to open less dialogs than in the other conditions when selecting a modality. This presumably led to the advantage of this approach in comparison to the other conditions.

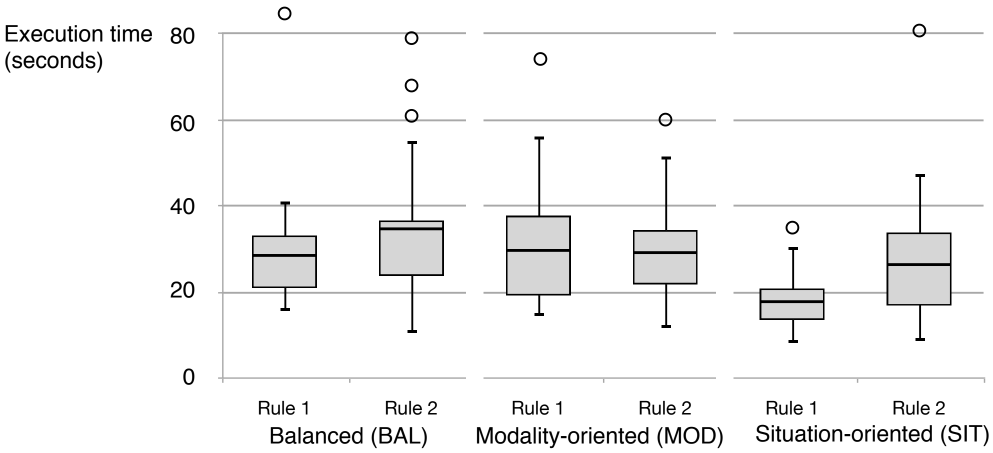


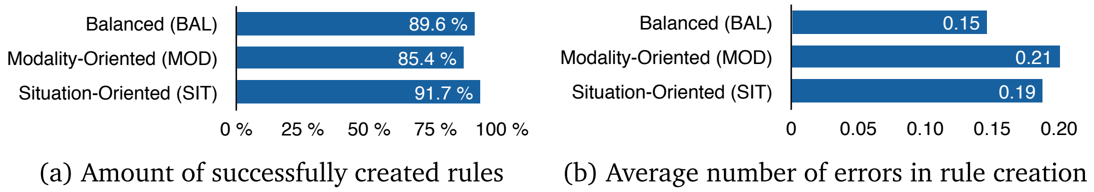
Figure 6.12: Rule creation time in the different conditions *SIT*, *MOD*, and *BAL*. The bars inside boxes indicate mean values. Outliers are marked as circles.

RQ2: Effectiveness

With all approaches, subjects were mostly able to create the demanded rules successfully. Success rates lay between 85.4% and 91.7% (see Figure 6.13a). With a repeated-measures ANOVA, we did not find a significant difference between the approaches (F(2, 46) = 0.46, p = 0.64, partial $\eta^2 = 0.019$). Subjects also made only few errors: The average number of errors lay between 0.15 and 0.21 in rule creation processes (repeated-measures ANOVA showed no significant difference between approaches, F(2, 46) = 0.31, p = 0.74, partial $\eta^2 = 0.013$). The results can be seen in Figure 6.13b. As most frequent error type, subjects chose the wrong conjunction (AND instead of OR). Further errors were made with modality states (e.g., vibration was set to ON instead to OFF) and with context factors (e.g., another than the requested ambient noise level value was set, or unnecessary conditions were included to the rule). We classify the majority of the errors made as slips (especially the ON/OFF and OR/AND confusion), rather than as errors resulting from a lack of understanding.

RQ3: Satisfaction

We found a significant effect of approaches on satisfaction measured by SUS (two-way repeated-measures ANOVA; F(2, 46) = 4.57, $p < 0.05$, partial $\eta^2 = 0.166$). The *SIT* approach received a SUS score of 86.7 (corresponding to excellent usability according to [15]). This is significantly better than *MOD* (74.9), as shown in a post-hoc t-test with Bonferroni correction ($p < 0.05$). The difference to *BAL* (78.4) was not significant.



(a) Amount of successfully created rules (b) Average number of errors in rule creation

Figure 6.13: Success (a) and error rate (b) of rule creation in the conditions *SIT*, *MOD*, and *BAL*.

Furthermore, subjects rated ease of use and clarity on a 5-point Likert scale after each method. In ease of use, *SIT* was again rated best with averagely 4.4, followed by *MOD* and *BAL* with each 4.0. Likewise, *SIT* received a rating of 4.4 in clarity, while the other approaches were rated with 4.1 (*MOD*) and 3.9 (*BAL*). With two-way repeated-measures ANOVA, we found a significant effect on clarity ($F(2, 46) = 3.52$, $p < 0.05$, partial $\eta^2 = 0.133$). There was no significant effect on ease of use ($F(2, 46) = 2.55$, $p = 0.09$, partial $\eta^2 = 0.100$) and of rule types. With post-hoc Bonferroni-corrected t-tests, we found the significant difference of clarity between the *SIT* and *BAL* approach.

RQ4: Semi-Automatic Rule-Creation Approaches

Subjects expressed their opinion on the alternative approaches *Suggestion* and *Snapshot* which were presented in Section 6.4.1, and were asked whether they would like to use them.

79% of participants would accept the *Suggestion* concept. They liked the reduced workload to create rules, compared to the manual approach. Further positive aspects were that the suggested rules serve as inspiration, and that they help finding rules for recurring situations users were not actively aware of. Those who did not favor the approach found the suggestions unnecessary, preferred creating their own rules, were skeptical that the suggestion mechanism would work well, or did not want the application record usage behavior in the background.

Only half of the participants would like to use the *Snapshot* concept, which is clearly less than *Suggestion*. On the positive side, some subjects preferred removing unwanted context factors and triggers from a rule instead of adding the wanted ones. One participant pointed out that he liked the approach because *"it does not require any machine learning but still behaves intelligently"* in determining the current situation. On the negative side, subjects were overwhelmed by the amount of information and found it too complicated to use. They were especially irritated by context factors which they would not use frequently (e.g., ambient noise level). One solution for this could be to only include frequently used context factors or user-definable "favorites" in the snapshot.

RQ5: Participants' Usage Suggestions

We asked subjects if they could spontaneously name rules they would find useful in everyday life. 83% were able to name one; over 50% provided more than one suggestion. In the following, we briefly present and aggregate subjects' suggestions. Most frequently (50%), subjects proposed rules where passive modalities (e.g., mute or vibrate) are activated when the user arrives at the workplace or the university. Then followed the opposite category of rules (21%), i.e., enabling active modalities (e.g., ringtones) when returning home. This was justified by the fact that many subjects forgot re-activating sound after they had muted their device. Another 21% of suggested rules comprised active modalities with outdoor activities (e.g., walking or biking). Furthermore, individual subjects named rules including adaptations to loud or bright environments. One participant suggested reducing screen brightness at a low battery level. Another subjects propose that the device could be muted when turned upside down (similar to the "flip to mute" example we presented in Section 6.3.4). We can summarize that a majority of created rules corresponds to similar use cases, which is muting and unmuting the device between workplace and home/university. However, the remaining rules were very diverse and covered almost the entire range of context factors and modalities integrated in our prototype.

In addition, we inquired which other context factors (independent of their technical realization) subjects could imagine. We received interesting suggestions, like modality switches based on calendar events, the recognition of nearby persons of one's own social circle and according modality change, and categorization of locations (i.e., muting the device not only at a predefined place, but by category like "cinema" or "restaurant").

In order to give subjects an idea for input modalities, we had them try the "Mimikry Input" examples presented in Section 6.3.4 to launch selected apps. The apps were started using a WOz approach to be able to also realize context factors that were not implemented at this time (e.g., motion gestures). After the trial, we asked subjects for own ideas for novel input modalities, and asked them to realize these with the presented rule creation interface.

Subjects here came up with screen gestures, motion gestures, and button input. Screen gestures were mainly used to launch applications. For example, subjects proposed to draw letters (such as N for notes or B for the browser) or symbols (such as a checkmark for the ToDo application) on the screen as touch gesture. Gestures were also proposed for changing settings, e.g., the volume by an up or down swipe. As an example for a motion gesture, one subject had the idea to rapidly move the phone down (as if it would "fall" down) to launch the messenger application. Finally, buttons were suggested as shortcuts to frequently used actions, such as enabling vibration or the screen rotation lock.

In total, subjects very much appreciated the presented system. On a five-point scale, 63% indicated to like the approach "very much"; the remainder of 38% stated to "like" it. 54% rated the system as "very useful", 42% as "useful", and 4% gave a "neutral" rating.

To further investigate the creation and application of rules under real-world conditions, we additionally conducted a field study, which we describe in the next section.

6.4.3 Field Study: Acceptance and Usage Patterns

In addition to the laboratory study described above, we were interested in usage patterns of our application, as well as in user acceptance under real-world conditions. To this end, we conducted a long-term study in the field.

Research Questions

The study has an explorative character. We investigated the following research questions:

RQ1 How many and which rules do subjects create?

We are here interested in the context types and modalities subjects integrate into rules, and in the complexity of rules that subjects create. In combination with the satisfaction ratings (see RQ4), this gives us an indication if the complexity and supported nesting depth of our system is sufficient for the use cases subjects want to cover.

RQ2 How important are different contexts and modalities to subjects?

Besides the information which contexts and modalities are actually used in rules, we also want to know which ones are most significant to users. While RQ1 is based on log measurements, this question is investigated through user feedback.

RQ3 How reliable does the system work?

Reliability is determined based on users' ratings. It depends on whether the defined rules lead to the expected behavior in the respective situations, or the system's output did not comply with users' expectations.

RQ4 How satisfied are users with the system?

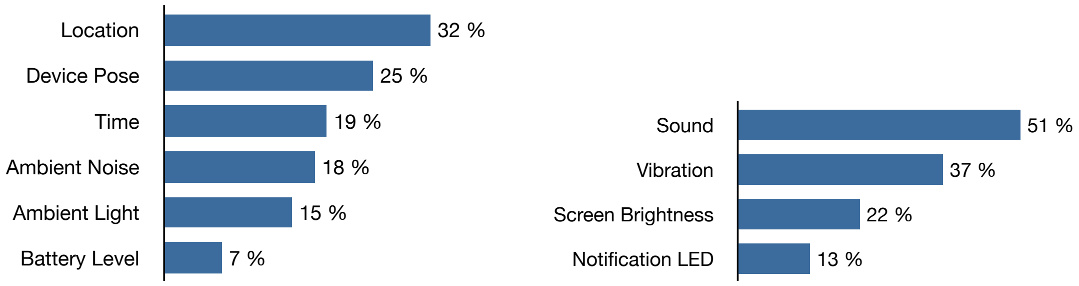
This question refers to overall satisfaction with the application, both in terms of usability and of capability, i.e., if it allowed subjects to do everything they wanted.

Method

Subjects were instructed to install the tasking application as described in Section 6.3.4 and use it to create rules, which they would use over the study period of two weeks. The app employed the *Balanced* layout variant, since we did not want to urge users into situation-oriented or modality-oriented thinking. In an introductory email, we explained the application to the subjects and showed them some example rules. Subjects were encouraged to actively try the application, explore which rules could be created, and pay attention if these rules were applied correctly in everyday use. However, there was no obligation or minimum usage requirement. Once a day, users were asked to answer a short questionnaire about the preceding 24 hours. In these questionnaires, subjects had to state how reliable the app has worked in that period, how satisfied they had been with the app's functionality, and if they had encountered any problems. This regularly shown questionnaire was realized by the SERENA self-reporting application (see Section 7.3 for an extensive description). Moreover, SERENA was used to log data on the created rules, such as the number of rules created, and the therefore integrated contexts and modalities. At the end of the study, subjects filled out a final online questionnaire.

Participants

Due to the explorative character of the field study, it was conducted with a smaller number of participants than the previously described laboratory study. Five participants (4 males, 1 female) took part; they were aged between 23 and 31 years (M = 27, SD = 3). Three of them were students, one was a software developer and one a physician. Subjects rated their level of expertise with smartphone with averagely 4.6 (SD =0.9) on a scale from 1 (= beginner) to 5



(a) Context factors included in rules

(b) Modalities adjusted by rules

Figure 6.14: Composition of rules created by participants in the field study. The percentage values indicate the amount of rules in which the respective context factor or modality was contained.

(= expert). All participants used their personal Android smartphones on which they installed the tasking application (3 had a LG Nexus 5, one owned a LG Nexus 4, and one a Sony XPeria Z1).

Results

RQ1: Created Rules

In total, subjects created 73 rules during the two-week study period. This corresponds to an average of 14.6 rules that each participant created, which indicates that subjects were eager to experiment with the system. 35 rules were deleted; 33 were disabled or modified in the course of the time. These numbers show that subjects made use of the possibility to manage and edit their list of active rules after initial creation. In particular, the fact that they often preferred to disable rules, without deleting them entirely, indicates that users prefer different sets of active rules depending on the situation.

The average complexity of rules was low: A rule comprised averagely 1.17 context factors and 1.13 modalities. Thus, most rules apparently were of a simple "if-then" structure, or contained maximally one AND/OR conjunction. Figure 6.14 gives an overview on employed context factors and on modalities that were adjusted by rules. The most frequently used context factors were location and orientation, which were part of 32% and 25% of all created rules. Orientation here refers to the device pose and was presumably used for "device gesture" rules. Time was used in 19% of rules, followed by ambient noise (18%), ambient light (15%) and battery level (7%). On output side, the most frequently influenced modality was sound. It was adjusted by 51% of all created rules. Further, vibration was adjusted in 37% of rules, followed by brightness (22%) and the notification LED (13%).

RQ2: Importance of Contexts and Modalities

Subjects rated on a 5-point Likert scale how important the availability of different context factors and modalities was to them. We found a high correlation with the context factors and modalities that were preferably used in rules (see RQ1). Device pose, time and location were most significant to users: Three to four out of five subjects responded that these context factors are “important” or “very important”. No subject at all considered location as “not important”. Less significant were battery, ambient light, and ambient noise: The latter was not considered as important by any of the participants. Interestingly, this contradicts the fact that 18% of all rules contained ambient noise as context factor. Probably, these rules were created just for testing purposes, but not used in practice later.

The most important modalities to be adjusted were vibration (rated as “important”/“very important” by four subjects) and sound (three “important”/“very important” ratings). Notification light and screen brightness seemed less interesting to subjects; only one considered these “important”.

RQ3: Reliability

On a daily basis, subjects stated in a pop-up questionnaire whether they were satisfied with the reliability of rules that day (i.e., each participant answered this question 14 times over the course of the study). We received in 51% the answer “very reliable” and in 46% the answer “mostly reliable”. Only 3% of all answers reported a “not reliable” application of rules. This result is very satisfying, confirming that the system could correctly determine the defined contexts and adapt modalities according to users’ expectations.

RQ4: Satisfaction

Similar to the reliability, subjects indicated their overall satisfaction once a day. In total, we received in 41% of all responses the statement that subjects were “very satisfied”; in 54%, they were “mostly satisfied”. In only 4% of all answers, subjects reported to be “not satisfied”. These high satisfaction ratings correspond with the feeling of reliability determined in RQ3. At the end of the study, the overall satisfaction with regard to usability was rated by a SUS questionnaire [34]. The system received a score of 75.0, which is a satisfactory result (according to [15], values between 71.4 and 85.5 correspond to “good” usability). In free text responses, some subjects stated that the battery performance was negatively affected while using the system, which is one point for optimization in the future. Moreover, two subjects would like to have more sophisticated options for the location context factor. Here, it should be possible to define a custom radius for geofences, and to search for location names instead of manually positioning a pinpoint.

6.4.4 Discussion and Lessons Learned

We now summarize the lessons learned and limitation of the conducted studies.

Laboratory Study

The “winner” of the laboratory study was the situation-oriented (*SIT*) rule creation interface. Subjects were significantly faster using this approach and also preferred it regarding clarity and usability, as measured by SUS (together with *BAL*). Regarding effectiveness, there was no significant difference between the approaches.

What is the reason for the advantage of the *SIT* approach? Presumably, subjects found situation-oriented thinking most intuitive, compared to the other approaches. After analyzing the video recordings, we believe that the compactness of the layout, compared to the other UIs, may be a further responsible factor. With *SIT*, users were able to view all modality controls at a glance (see Figure 6.10b), and less scrolling was required. Furthermore, less dialogs had to be opened and settings could be directly made from within the main screen.

Field Study

Our field study was conducted with a small number of participants and for a rather short period of two weeks. The duration was set in concordance with the findings in Section 7.4.3, where we learned that the study duration should be limited if additional tasks (such as regular questionnaires) are demanded from users. In order to identify long-term effects, such as a potential rise of rule complexity over time, an extended study duration would be necessary as future work. Likewise, a higher number of participants could produce a more diverse set of rules created, as well as statistically significant usage patterns.

However, this actual study served as proof-of-concept for our system in practice, and as a first indicator for reliability and acceptance. We received to a great extent positive feedback, proving the applicability in real-world conditions. Subjects were also satisfied with the range of functions, indicating that there is no urgent demand to create more complex rules. This confirms us in our decision to refrain from a higher nesting depth when creating logical expressions, for the sake of a simpler UI (cf. the discussion of this aspect in Section 6.4.1). Further, it justifies our decision to use a rule-based approach, rather than, e.g., machine learning.

As the rule evaluator is constantly running as a background service, our system affects the battery life of the smartphone, which was also noticed by some participants in this study. Optimization with regard to energy consumption was not a focus in this stage of the implementation. However, we are aware that this aspect must be taken into consideration for a productive use, as it may influence the users' general acceptance of the system. In the underlying M3I framework, we already offer the possibility to define rule-specific intervals for how often the rule is evaluated, so that rules that are not time-critical can be implemented in an energy-conservative way. However, the ability to set this interval was not included in the GUI for the field study for reasons of simplicity.

A final lesson learned from subjects' feedback is that they desired to be notified when a rule was applied. In the system used in the field study, subjects could only tell from the built-in indicators (e.g., for the volume) that a modality setting had been changed. Explicit methods to notify the user would be, e.g., alerts, widgets, or system notification messages. Consequently, in the subsequent section, we present and evaluate different concepts for increasing awareness on automated modality changes.

6.5 User Interfaces to Achieve Modality Awareness

We first propose three alternative concepts for awareness on output modalities, which were partly informed by the initial focus group discussion (see Section 6.2.3). Subsequently, we report on the conducted laboratory study and the obtained results.

6.5.1 Concepts

We present the methods *Widget*, *Alert*, and *Notification* to inform the user on which (output) modalities are currently active, and whether a rule-based modality change has occurred. Each concept can be realized in two manifestations: opt-in and opt-out. With opt-in, the user is informed on a potential modality change, but must actively confirm it. If no action is performed, the old state is maintained. With opt-out, the modality change is applied automatically if no action is performed, but the user has the possibility to revert the change. While opt-out is adequate for a high level of trust that the automated rule application process works as desired, opt-in minimizes the risk that an undesired modality change occurs.

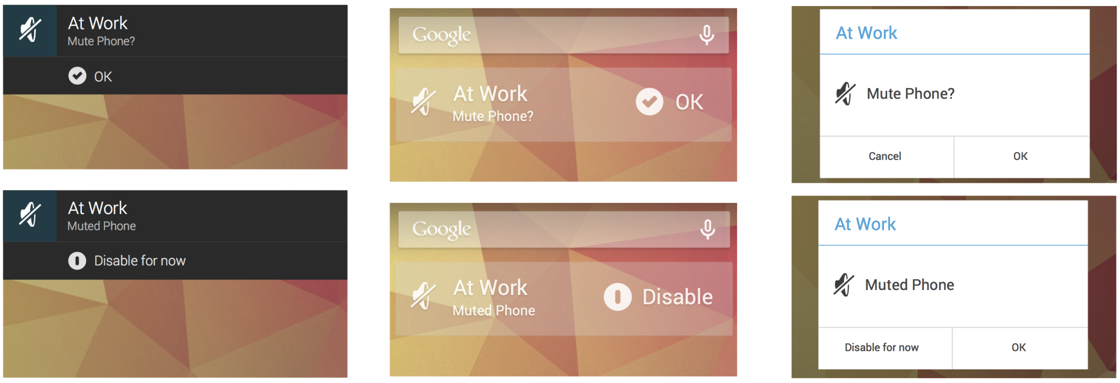
Notification A notification appears on the top of the screen, informing about the rule name and the modality change (see Figure 6.15a). Unlike the *Alert*, the notification only has one action button: In opt-in mode, there is an “OK” button to apply the rule; in opt-out mode, there is a “Disable for now” button to revert the change.

Widget A home screen widget permanently visualizes the currently active output modalities (see Figure 6.15b). It dynamically updates its content when a rule is applied and the modalities change. We consider this visualization as unobtrusive, as it does not interrupt the user’s workflow. However, this approach requires active checking of the widget’s state by the user.

Alert An alert message pops up when a modality change occurs (see Figure 6.15c). The message box shows the name of the applied rule, as well as the modality change to be performed. In opt-in mode, the user has the options “OK” to activate the change and “Cancel” to dismiss the dialog. In opt-out mode the rule is applied automatically, but the user can “Disable” it, or acknowledge the change and dismiss the dialog.

These three concepts represent different compromises regarding level of control and obtrusiveness. The *Alert* provides a very high amount of control of modality changes, but is most obtrusive of all visualizations. As it is modal, a user reaction is required in any case. *Widget* is the least obtrusive visualization, but it also provides the least amount of control: users must actively return to the home screen and peek at the widget’s state to be informed on modality changes. *Notification* is a compromise in both obtrusiveness and control. The visualization proactively informs on modality changes, but there is no need to react immediately.

Independently of the used visualization concept, there is always also the operation-system-specific symbol (on top of the screen, next to the time) from which the user can deduce the currently active modality.



(a) Notification (b) Widget (c) Alert

Figure 6.15: Visualization concepts for awareness on rule-based modality switching. The top images show the opt-in variants, the bottom images the opt-out variants.

6.5.2 Laboratory Study

Research Questions

RQ1 Which concept is most efficient?

We compare the concepts with relation to the time until subjects noticed the modality change.

RQ2 Which concept is most effective?

Similar to the laboratory study on rule creation, we look at success rate and error rate. As “success”, we consider the case that subjects did notice the modality change. “Errors” denote the situation when subjects performed an action other than the requested disabling of the modality change (e.g., click the wrong button in the *Alert*, or swipe the *Notification* away instead rejecting it).

RQ3 What are user preferences with regard to the approaches?

For this research question, we asked subjects which of the presented method they prefer, and why.

Method

Task

Each participant evaluated the awareness visualizations *Notification*, *Widget*, and *Alert*, each as opt-in and opt-out variant (see Section 6.5.1 for the description). Consequently, there were six conditions in a within-subjects design. The order of conditions was counterbalanced using a Latin square. For each visualization, the task was to disable (i.e., reject) the modality change as soon as noticed by the visualization.

All awareness visualizations are intended to be noticeable in a peripheral way, while the user is focused on another task on their phone. Therefore, we engaged subjects in a primary task that required their attention. At a random time, the experimenter triggered the notification (using a WOz application running on a laptop).

As primary task, subjects were asked to search for a specific photo using a photo viewer widget. The widget allowed to browse through a list of photos by swiping, where one photo was shown at a time. To ensure that the subject remained engaged in the primary task, the task was designed as "unsolvable", as we gave the instruction to find an image (showing a giraffe) that was not contained in the collection.

Measurements

From the moment the visualization was triggered, the time until the subject clicked the notification, widget, or alert was measured. This measurement was done automatically by the prototype application and used to determine efficiency (RQ1). As reasonable interval in which the subject should have noticed the modality change, we defined a timeframe of 15 seconds after the notification was triggered. If the subject did not notice the change within this interval, we stopped the trial and classified it as unsuccessful. Effectiveness (RQ2, success rate and errors) were detected by analyzing a video recording of the task. User preferences (RQ3) were determined by a questionnaire after the hands-on part of the study.

Participants

The study was conducted with 24 participants (6 females, 18 males), aged between 20 and 31 years (M = 24, SD = 3)[85].

Results and Discussion

RQ1: Efficiency

With a two-way repeated-measures ANOVA, we found a significant effect of visualizations on task completion time (F(2, 46) = 28.34, $p < 0.001$, partial $\eta^2 = 0.552$). *Alert* was significantly faster than the other conditions with averagely 2.1 s ($p < 0.001$ in a post-hoc Bonferroni-corrected t-test). The average times with *Widget* were 7.6 s and with *Notification* 9.9 s. Figure 6.16 visualizes the results.

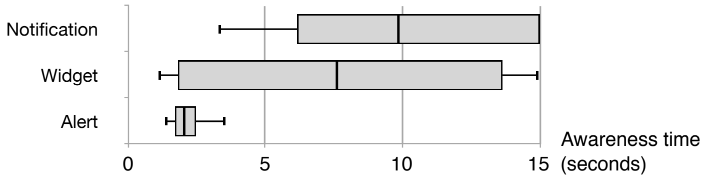


Figure 6.16: Time until subjects became aware of modality changes in the different conditions *Notification*, *Widget*, and *Alert*. The bars inside boxes indicate mean values, the whiskers minima and maxima.

[85] The participants were the same as in the laboratory study described in Section 6.4.2, as both studies were conducted in common. Since both studies investigated different systems, and within-subjects-designs were used, no influential or priming effects are expected.

RQ2: Effectiveness

There was a significant effect of visualizations on success rate (two-way repeated-measures ANOVA; F(2, 46) = 13.36, $p < 0.0001$, partial $\eta^2 = 0.367$). Success rates were best for *Alert* (100%) and *Widget* (88%), showing no significant difference in a post-hoc t-test with Bonferroni correction. With *Notification*, the success rate was with 54% significantly worse than in the other conditions ($p < 0.05$). The error rates ranged between 0.0 (*Widget*) and 0.1 (*Notification*), and did not significantly differ between conditions (F(2, 46) = 1.53, p = 0.23, partial $\eta^2 = 0.062$)

RQ3: User Preferences

Subjects indicated which visualization the like best, where they could choose between *Notification*, *Widget*, and *Alert*, each in opt-in and opt-out variant, and additionally the option of no visualization at all. The results are summarized in Figure 6.17. The most popular visualization was clearly *Notification* with 42% (thereof 38% as opt-out variant). 25% of subjects voted for *Widget* (all of them as opt-out variant). *Alert* was only preferred by 21% (13% as opt-out and 8% as opt-in). Another 13% would prefer no awareness visualization at all.

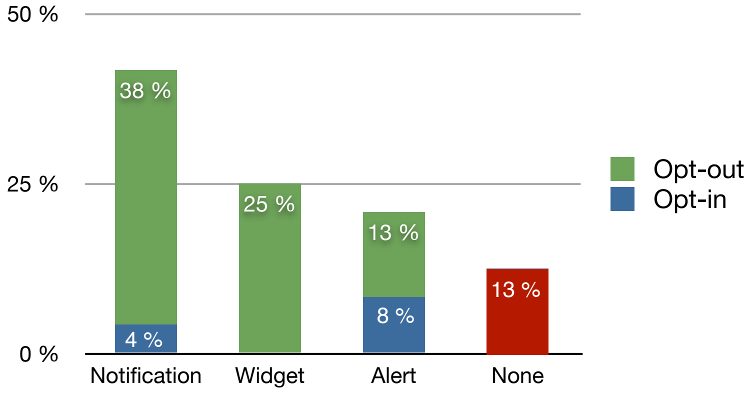


Figure 6.17: User preferences for modality awareness visualizations

6.5.3 Discussion and Lessons Learned

The joint results of the quantitative measures and subjects' preferences form an interesting picture: Although *Notifications* were clearly the most popular visualization, they showed a significant lower success rate and the slowest notification time (i.e., worst efficiency). By contrast, the *Alert* visualization, which was best both in success rate and efficiency, scored last in user preferences. A reason can be found in subjects' oral explanations of their preference voting. The prevalent reason for voting for *Notification* was the appropriate level of obtrusiveness. The *Alert* was perceived as too distractive in the middle of a task by many participants. This shows that obtrusiveness was more significant to users than efficiency. A second explanation could be that, although we found significant effects of visualizations on efficiency and effectiveness, none of the visualizations was actually really bad. They showed almost no error rates, even *Notifications* were still noticed in averagely less than 10 seconds, which seemed to be sufficient for most subjects.

The short awareness time of *Widget* might partly due to the primary task of subjects in this study, which involved another home screen widget. This allowed subjects to glance at the awareness widget at the same time. If subjects had been using a full-screen application, the *Widget* condition would probably have yielded worse efficiency results.

The opt-out variants were generally preferred more (76%) than opt-in (10%). This indicates a relatively high confidence in automatic modality switching. Subjects accept rather rejecting individual modality switches, instead of having to acknowledge every single rule.

6.6 Summary

In this chapter, we have presented a rule-based multimodality framework and several user interface concepts for programming multimodal behavior and for awareness on multimodality changes. We conducted comparative evaluations in the laboratory and evaluated rule creation in the field.

In summary, we gained the following findings and recommendations:

- Subjects show a generally high interest in proactive modality switches, allowing automatic adaptation of modality settings to different contexts using a rule-based approach.
- Subjects are generally confident in automated, rule-based modality switching. The majority preferred an opt-out over an opt-in system.
- Most rules created by subjects are of a simple “if-then” structure and related to location contexts. The prevalent output modalities influenced by rules are sound and vibration.
- The rule creation interface that achieved best results in terms of efficiency and usability is the situation-oriented variant.
- The recommended awareness visualization, although slower in efficiency, is the one using opt-out notifications, due to their low level of obtrusiveness.
- Nevertheless, user preferences are heterogeneous, which should be respected by the possibility for custom settings (e.g., allowing a choice between alternative visualizations and opt-out/opt-in version, possibly even on a per-rule basis).

Chapter 7

Evaluating Mobile Multimodal Systems

7.1 Problem Statement and Research Questions

Evaluation is a crucial part in the development process of systems and applications. After we have discussed the design and implementation of multimodal and sensor-driven applications in the previous chapter, this last big chapter of this dissertation is devoted to the evaluation process.

In Section 2.4, we gave an overview on evaluation methods that are candidates to be used in conjunction with MUSED systems. Each of these evaluation methods have, however, individual strengths and weaknesses. Their suitability also depends on the kind of MUSED system, the stage of maturity in the development process, or the aspect to be evaluated.

The high-level research question treated in this chapter is:

- Based on the characteristics of MUSED systems described earlier, what are the implications for appropriate evaluation methods?

We now present selected evaluation methods in detail, informed by our experiences from their application in our research. In Sections 7.2 and 7.3, we report on laboratory evaluation (at the example of Wizard-of-Oz (WOz) testing) and real-world evaluation (at the example of SERENA, a self-developed logging and experience sampling tool). Furthermore, we discuss two special cases, especially relevant for MUSED systems: long-term evaluation (Section 7.4), and evaluation in the large (Section 7.5). We have successfully employed all of these methods with the prototypes presented in Chapters 3–5, and demonstrated their applicability in a broad variety of application areas. This allows us to report on the advantages and lessons learned in the individual research projects. At the end of the chapter, we give recommendations for evaluation methods throughout the development process of MUSED systems.

This chapter is partly based on two papers we have published in 2012 and 2013 [228, 230].

7.2 Laboratory Evaluation: A Case for Wizard-of-Oz Testing

In this section, we motivate Wizard of Oz (WOz) testing for laboratory evaluation of multimodal user interfaces and interaction techniques. As outlined in Section 2.4, the WOz technique denotes the simulation of (parts of) the actions and/or reactions of a system by a human acting as “wizard” [152].

We promote the use of WOz testing in the field of mobile multimodal interaction for several reasons.

- Creating fully-functional novel multimodal and sensor-driven interfaces often involves considerable implementation effort. The implementation is often more sophisticated or contains more complex algorithms than it is the case with conventional user interfaces. However, frequent research questions of interest are acceptance, usability, likability, etc. These can be examined independently of the algorithms, techniques, platforms, frameworks, and APIs used for the implementation. With WOz testing, these research questions can be investigated without having to fully implement the various interfaces.
- The conceptual evaluation of interaction methods shall be conducted *early* in the design process to inform further development. Often, it is not sufficient to conduct such evaluations based on descriptions, images or videos. Instead, *interactive* prototypes are required for subjects being able to evaluate a novel interface comprising multimodal components. This interactive component can be added by the WOz approach.
- One might want to conduct comparative user studies with different conditions, where internal or environmental factors are varied. WOz testing can help simulating and reproducing such conditions for reliably repeating experiments under exactly the same conditions, without the need of implementing all variants.
- Immature or buggy implementations can decrease the perceived utility, usability and likability, and thereby distort results. Given the fact that a user study never evaluates an interaction technique as such, but always in the context of a particular implementation, WOz testing helps isolate these factors as good as possible, thereby adding to more valid results.

We pursued this approach in several studies described in this thesis, e.g., in the context of indoor navigation (see Sections 5.3.3 and 5.4.2), modality awareness interfaces (see Section 6.5.2) and multimodal input methods (see Section 6.4.2). In the following, we summarize the advantages we obtained through using WOz for these studies.

Example 1: A Prototyping Tool for Indoor Navigation User Interfaces

We simplified and accelerated the evaluation of user interfaces for visual indoor navigation system by using WOz testing.

- We saved time to set up the underlying localization mechanism, which would involve the creation of a reference image database and the integration and fine-tuning of computer vision algorithms.
- We could evaluate different UI concepts even before the live routing algorithm was implemented and available as fully functional service.
- The evaluation could be conducted at any location, in particular at other locations than the final deployment area. Testing with an underlying live localization system would have required the creation of an entirely new reference database, setup, etc., for each new location.

- We can simulate different levels of inaccuracy (of the location or orientation estimate, or both together) in order to find out how the user can deal with this inaccuracy using a particular visualization. This amount of control over localization accuracy would not be possible with a live system, as localization accuracy is influenced by multiple external factors, e.g., lighting conditions.
- We can investigate the (motivational or affordance) effect of individual user interface elements, such as the indicators motivating users to point at feature-rich regions, and rapidly iterate on their design based on our findings.

For more details, see Sections 5.3.3 and 5.4.2.

Example 2: A Tool for Prototyping Multimodal Input Methods

We demonstrated various multimodal interaction methods to subjects and used WOz to make these interaction methods work.

- Interaction modalities that would be challenging to implement (e.g., including motion gestures) could nevertheless be used to show the design space and to collect feedback.
- As the presentation of modalities played only a minor role in relation to the entire laboratory study, they could be included without spending too much time for their implementation.
- The case that subjects dislike an interaction modality because it does not properly work could be excluded. Our goal was a conceptual demonstration, not an evaluation how well a modality actually works.

For more details, see RQ5 in Section 6.4.2.

Example 3: Illustrating Modality Awareness Visualizations

We simulated modality changes to test the efficiency and effectiveness of different UIs to maintain modality awareness.

- In a live system, modality switches would in the real world occur based on (context-specific) rules. In order to trigger these switches, subjects would have to change their physical location, or the context (e.g., ambient light, noise) would have to change. Since this is impractical for a laboratory study, we triggered the modality changes independently of the context. WOz works here as simulation environment.
- Context changes, as they would be necessary to trigger modality changes if we did not use WOz, are a potentially distracting factor. As they cannot be fully controlled, they could distort efficiency and effectiveness measurements. With WOz, we eliminated external influences as good as possible.

For more details, see Section 6.5.2.

7.3 Real-World Evaluation: SERENA – A Framework for Logging and Self-Reporting

Laboratory studies can provide initial feedback on a system that informs further design, or investigate an individual, tightly focused usability question, e.g., the comparison of two interaction types or interfaces. However, for many research questions in HCI, user studies in the lab are not sufficient. This is especially true for MUSED systems, as for these systems:

- the user interface is often determined by contextual information [224, 225]
- the environment is often explicitly involved in the interaction process [168, 225]
- modality usage in certain social settings shall be investigated, e.g., in a meeting, in the library, in a restaurant, at home [147, 275]
- modality usage in certain states or modes of transportation shall be investigated, e.g., while walking or driving [147]

Such questions can only be evaluated "in the wild". Moreover, researchers are often interested in usage patterns, adaptation processes and learning curves. This must be investigated in long-term studies observing the users' behavior over weeks, months, or even years.

We developed a toolbox supporting data collection in long-term studies that we called SERENA (SElf-Reporting and ExperieNce sampling Assistant). SERENA is available for download at `http://www.eislab.net/serena`. Even though there exist various data collection applications in research (see Section 2.4), none of these applications gave us a convincing picture of the whole survey process.

7.3.1 Functionality

SERENA combines questionnaire-based self-reporting with automated data logging in a singular smartphone application. It is designed flexible enough for a variety of use cases and can be customized to the experimental needs. We successfully employed SERENA for multiple studies we have conducted within the research for this dissertation. SERENA has been used:

- to capture long-term usage patterns of the MobiDics didactics tool (application usage logging), described in Section 4.3.3
- to understand everyday usage of rules for multimodal behavior of mobile phones (used for diary-based self-reports), described in Section 6.4.3
- to investigate the reliability of self-reporting in comparison to logging (used for experience sampling, diary, and logging), described in Section 7.4

These examples illustrate the diverse possibilities to apply SERENA. In the following, we detail the possibilities that SERENA provides.

Logging

SERENA can log when applications are started and, within applications, the on-top (Android) Activity, i.e., screen page. Furthermore, location information (from the most accurate active location provider) can be stored with each log entry. In a configuration file, the experimenter can specify a list of applications or activities that should be under observation by SERENA. That way, surveillance can be confined to applications that are relevant for the study, with the goal to preserve the privacy of subjects as much as possible. With location logging, app usage can be monitored along location traces and, based on location clustering, associated to frequent place such as "home" or "work". By this means, research questions like *"do usage patterns change between different social contexts"* can be tackled.

Self-Reporting

SERENA supports self-reporting based on questionnaires, utilizing the following question types:

- **Single Choice**: A singular answer can be selected from a list of possibilities using radio buttons.
- **Multiple Choice**: Multiple answers can be selected from a list of possibilities using checkboxes.
- **Drop-Down**: A singular answer can be selected from a list which appears in a drop-down menu. While the single choice type is adequate only for a limited number of answer possibilities, the drop-down menu can contain a large number of elements.
- **Likert**: Subjects can indicate their level of agreement to a statement on a scale with a defined number of steps (e.g., five or seven).
- **Free Text**: A (single-line or multi-line) text field is provided, where the question can be answered in free text.
- **Range**: A value of a defined range (integer or float) can be specified using a slider.
- **Instruction Text**: In addition, a text-only page can be used for instructions and messages (such as an introductory page or a "thank you" page at the end of the questionnaire).

Questionnaires can be configured to be *interval-based*, *event-based*, or *voluntary*. With interval-based questionnaires, e.g., appearing once a day, diary-like studies can be realized. Event-based questionnaires are triggered by the usage of certain apps. For example, SERENA can be configured to show a survey after a participant has used the prototype application that shall be evaluated in a long-term study. Voluntary questionnaires are opened manually by the participant and can be filled out any time.

The experimenter can assign a group ID to each questionnaire and specify in which timespan it is valid (e.g., a two-week interval). With these features, SERENA supports multiple conditions in within-subjects and between-subjects study designs. A within-subjects study can be realized by multiple questionnaires with different timespans. For example, participants receive questionnaire A in the first part and questionnaire B in the second part of the study. For a between-subjects study, different group IDs can be assigned to questionnaires. Group 1 could then, in parallel, receive questionnaire A, and group 2 could receive questionnaire B.

Remote Operation

SERENA can be controlled and deployed remotely using a web application. Researchers can configure all settings (e.g., which applications should be logged, query intervals for questionnaires), create the questionnaires in the web application, and subsequently generate a customized Android application to be distributed to participants.

The participants install this application on their smartphone at the beginning of the experiment. The collected data (usage logs and answers to questionnaires) are automatically uploaded to a server. Even if the application is already deployed, settings can be altered and new questionnaires can be sent remotely to participants' devices. Participants do not need to come to the lab for application setup or for collecting the data from their devices at the end of the experiment, which greatly simplifies setting up studies. We emphasize that we discuss here only technical requirements; a personal meet-up with subjects might still be required for signing a consent form.

7.3.2 Implementation

Backend

The backend fulfills two main tasks: the configuration of SERENA prior to the experiment, and the analysis of the collected data. The configuration tool allows to design questionnaires (see Figure 7.1a) which can be exported as XML file. This file also contains configuration data (i.e., query frequency for questionnaires, information which data should be logged, etc.). With this file, the researcher can compile a customized version of the SERENA Android application that can be distributed to participants. Alternatively, instances of SERENA already installed on the participants' devices can remotely be updated with new questionnaires.

The backend is implemented with the Python Pyramid framework[86]. The web pages for creating questionnaires and analyzing the results are created using jQuery[87] and Ember.js[88]. Logs and survey data received from subjects' phones are saved to a MySQL database. Using the analysis webpage of the backend, the data can be reviewed and exported for in-depth statistical analysis.

Mobile Application

The mobile SERENA app is implemented in Android (version 2.3). Its core is a background service which monitors device usage and schedules questionnaires. Device usage is logged

[86] http://www.pylonsproject.org/projects/pyramid/about, accessed June 29, 2014
[87] http://jquery.com, accessed June 29, 2014
[88] http://emberjs.com, accessed June 29, 2014

according to the specified filters by the researcher (see Section 7.3.1). Log data are regularly uploaded to the server and additionally saved locally on the device. SERENA creates a unique ID for each installation and attaches it to log data entries and answered questionnaires. This allows researchers to match logged data and self-reports, while not revealing the identity of individual participants.

For scheduled questionnaires (in *interval*- or *event*-based mode), a notification message appears in the notification bar. The user can decide to immediately answer the questionnaire or to postpone it, in order to minimize disturbance in an ongoing task. Users can also manually bring up questionnaires from within the SERENA application. The main screen shows the currently available list of experience sampling questionnaires, where questionnaires that are added or removed by the experimenter during the study appear or disappear automatically.

7.4 Data Collection in Long-Term Studies

As we are, for the long-term evaluation of MUSED systems, often interested in user attitudes, adoption, and qualitative feedback, we look in detail at self-reporting in this section. The commonly used techniques like experience sampling or the diary method bear challenges in long-term use. Experience sampling can be highly interruptive and burdensome if the sampling rate is high, and the study is conducted over a long time. With diaries, by contrast, it may happen that subjects do not remember all events throughout the day if they, e.g., only make one entry every evening, or that they might refrain from writing down certain events, such as, e.g., intimidating information.

The reliability of self-reporting can be affected, e.g., by:

- the participants' self-perception (one's own behavior is perceived differently by oneself than from outside)
- memory (subjects may not remember correctly and forget events or actions),
- sluggishness or enervation (especially when reporting interrupts the current task),
- privacy demands (reporting can be embarrassing or make subjects feel to appear in a bad light; consider, e.g., media consumption, sports or food intake habits).

We conducted a structured exploration of how reliable self-reported information is under different conditions, in comparison to logged data as ground truth.

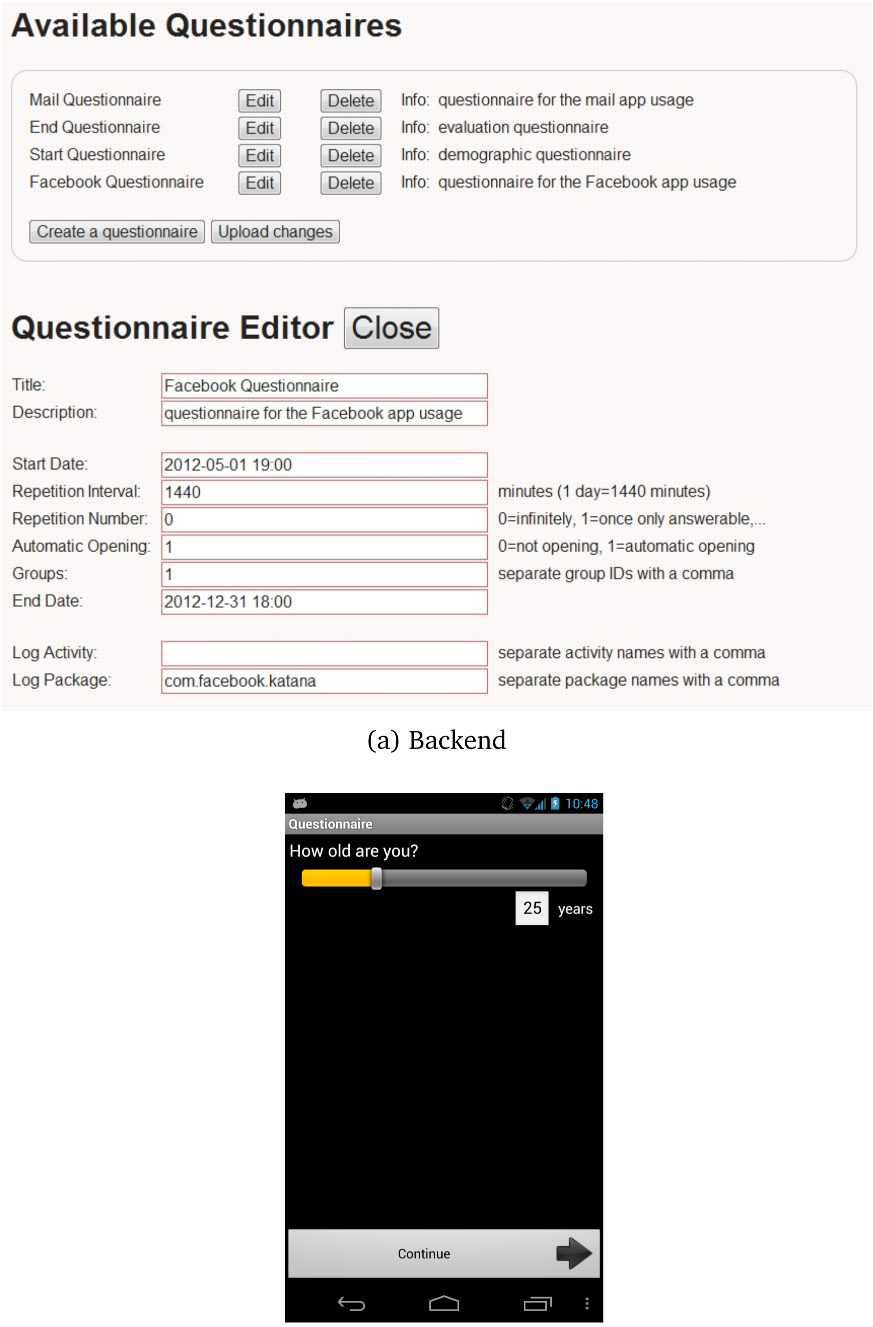
7.4.1 Research Questions

We analyze the following research questions.

RQ1 How reliable are self-reports in terms of the reported content?

RQ2 How reliable are self-reports in terms of reporting frequency?

RQ3 How reliable are self-reports over the course of a study?



Available Questionnaires

Mail Questionnaire	Edit	Delete	Info: questionnaire for the mail app usage
End Questionnaire	Edit	Delete	Info: evaluation questionnaire
Start Questionnaire	Edit	Delete	Info: demographic questionnaire
Facebook Questionnaire	Edit	Delete	Info: questionnaire for the Facebook app usage

Questionnaire Editor

Title:	Facebook Questionnaire	
Description:	questionnaire for the Facebook app usage	
Start Date:	2012-05-01 19:00	
Repetition Interval:	1440	minutes (1 day=1440 minutes)
Repetition Number:	0	0=infinitely, 1=once only answerable,...
Automatic Opening:	1	0=not opening, 1=automatic opening
Groups:	1	separate group IDs with a comma
End Date:	2012-12-31 18:00	
Log Activity:		separate activity names with a comma
Log Package:	com.facebook.katana	separate package names with a comma

(a) Backend

How old are you?

25 years

(b) Android application

Figure 7.1: Backend and mobile app of our self-reporting toolkit SERENA. (a) In the backend, questionnaires can be created, managed, and sent to the mobile app that subjects install on their smartphone during a long-term study. (b) The screenshot from the mobile app shows an example questionnaire that has been created with the backend.

These research questions address three aspects of reliability. First, do users accurately perceive and inform on their actions, or, e.g., misinterpret timespans (such as how long they used a certain application)? Second, do they accurately report occurrences of an action, or forget or deliberately conceal them (e.g., whether they used a certain application)? Third, besides overall accuracy, we are also interested in time-dependent effects: How does self-reporting behavior change in the course of the study? A reasonable hypothesis is that the commitment decreases over time.

RQ4 Do self-reports affect actual behavior?

By this research question, we investigate whether a study that requires subjects to report on their behavior reflects their behavior in the unobserved case, or if subjects alter their actual behavior because of the tasks imposed by the study.

RQ5 How frequently should self-reports be requested to still obtain accurate and useful results?

In order not to discourage participants, researchers might want to minimize the subjects' burden when participating in a study. This could be reached by reducing the frequency of self-reports. However, the cost might be that results become less accurate (see reasons above). Increasing the frequency might improve the results, but also exhaust users more. In order to answer this research question, we need to take into account multiple factors: Besides quantitative measurements of reporting accuracy in different conditions, also qualitative feedback and the perceived burden of users play a role.

7.4.2 Long-Term Study: Comparing Logging and Self-Reporting

Experimental Design and Task

In order to assess the reliability of self-reports, we needed a use case where ground truth data can be easily obtained. We therefore chose smartphone usage, as app usage information can be assessed well by logging (see, e.g., [19, 25, 74, 95, 145]). While SERENA is not limited to monitoring specific apps, and thus usable as research vehicle for various HCI studies on mobile devices, we constrained our analysis to two applications to limit the self-reporting burden for participants. As representative applications that are likely to be installed and regularly used by a majority of smartphone owners, we chose Facebook and Mail.

Subjects were instructed to answer a short questionnaire (self-report) each time they used either Facebook or Mail. Let us again note that self-reports were used only as vehicle to assess subjects' reporting behavior and not for in-depth usage analysis; therefore, we kept the questions simple. Subjects were asked to estimate how long they had just used Facebook or Mail, and how often they had used Facebook or Mail without filling out a questionnaire. The latter question gave subjects the opportunity to catch up on reporting, in case they had forgotten to answer a questionnaire. When a subject filled out a questionnaire directly after having used an app, we call this *direct self-report*. For example, a subject has opened Facebook three times and answered three questionnaires on those usages. When the subject only filled out a questionnaire after the third usage, indicating that she has used Facebook three times, the former two app usages are considered to be reported indirectly; we hence call them *indirect self-reports*.

The way how subjects were reminded to answer a questionnaire was varied in three “intensity levels”:

- In the *Voluntary* condition, users were never actively reminded to answer a questionnaire. The only given instruction was the initial task assignment prior to the study.
- In the *Interval* condition, a reminder notification appeared once a day (scheduled to 7:00 PM for Facebook, and to 9:30 AM for Mail). The reminder only showed up if reporting has been missed at least once since the previous reminder.
- In the *Event* condition, a questionnaire appeared right after each usage of the Facebook or Mail application. The questionnaire was, however, only shown if the application was quit with the home button and the user returned to the home screen. If the user had switched to another application using the multitasking view, or another application was started from within Facebook or Mail (e.g., when a mail attachment was tapped and opened in another application), no questionnaire was shown in order not to interrupt the task.

The study lasted six weeks. Participants were asked to install SERENA on their personal smartphone and to use the phone as usual. In the course of the six-week study period, a reminder email was sent every two weeks, thanking subjects for their participation so far and indicating how long the study would still last. In a pre- and post-questionnaire, we collected information on current smartphone usage (prior to the study), and to get feedback on the self-reporting process (after the study).

Participants

30 subjects (8 females, 22 males) participated in the study. They were aged between 18 and 32 years (M = 25, SD = 3) and recruited among acquaintances of the researchers. None were related to the research project. The only requirement to participants was that they were (Android) smartphone users and regularly used email and the Facebook on their device. Subjects received a small gift for compensation after the study. We deliberately decided for a modest compensation in order not to influence results through the incentive. We used a between-subjects design, with 10 participants in each of the conditions *Voluntary*, *Event*, and *Interval*.

Most subjects stated to use Facebook and Mail one or several times a day. Facebook was used several times by 21 subjects, once by six, and fewer by three participants. 25 subjects used Mail several times a day, two used it once, and three fewer than once a day.

Measurements

In all three conditions, SERENA logged participants’ actual usage of Facebook and Mail. If subjects switched back and forth between applications within a short period of time, we aggregated subsequent usages of the same app and counted them as singular app usage, summing up individual usage times. We assumed a singular task if users returned to the original app within 60 seconds. One example is when subjects were composing an email, looked something up in another app, and switched back to finish the email. Often, those other applications were launched from within the first application using an *Intent*, e.g., for choosing an email attachment, or sharing an image in Facebook.

Results

We first present the results of our individual measurements, before we put the results in relation to the formulated research questions and discuss implications.

Reliability: Number of App Usages

In total, 3,631 Mail usages and 3,181 Facebook usages were logged during the study. Figure 7.2 illustrates the ratios of self-reported app usages in relation to the logged usages.

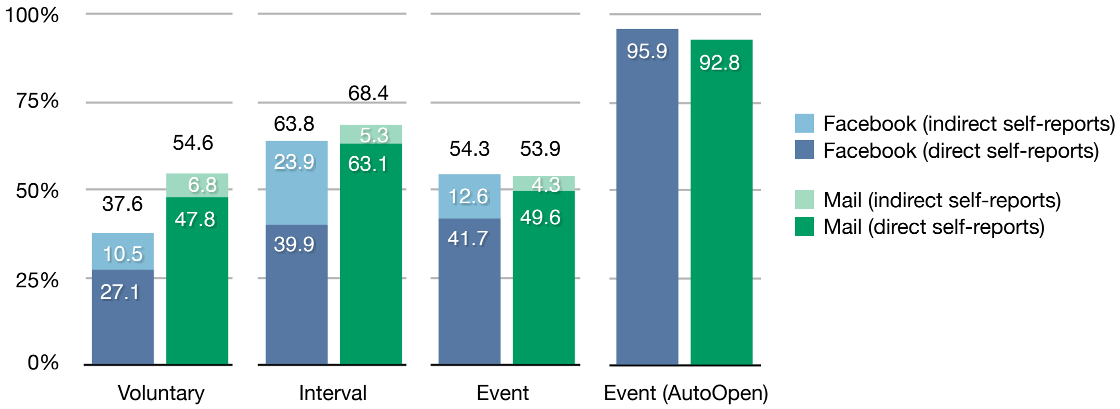


Figure 7.2: Ratio (in percentage) of self-reported and logged app usage in the conditions *Voluntary*, *Interval* and *Event*. *AutoOpen* denotes the amount of filled out questionnaires that have been opened automatically in the *Event* condition. Bars show the sum of direct self-reports (representing the number of filled out questionnaires) and indirect self-reports (which comprise app usages that have been caught up in a subsequent questionnaire).

The bottom, darker-colored parts of the columns represent direct reports (i.e., the actually filled out questionnaires). The top, light-colored portions illustrate indirect reports (see earlier explanation of the terms), so that the columns in total represent the amount of all reported usages. Subjects reported 37.6% of Facebook usages in *Voluntary*, 63.8% in *Interval* and 54.3% in *Event*. 54.6% of Mail usages were reported in *Voluntary*, 68.4% in *Interval* and 53.9% in *Event*. The differences were not significant between conditions ($p > 0.05$ in pairwise Student's t-tests).

As described in the *Experimental Design* section, in the *Event* condition, a questionnaire was only shown when subjects returned to the home screen after having used Facebook or Mail. This was the case in 1,224 of 2,488 app usages (49%). If we consider only these automatically shown questionnaires (we call them *AutoOpen*), the reporting ratios were 95.9% for Facebook and 92.8% for Mail. The reporting ratio in *AutoOpen* was significantly higher than in the other conditions ($p < 0.005$).

We can note the following observations.

- Even though the "intensity level" of reminders increased from *Voluntary* over *Interval* to *Event*, there was no effect on actually reported usages.
- If we only consider the reporting rate in *AutoOpen*, the *Event* condition actually lead to more reports than the other conditions.
- Facebook usages were more often indirectly reported than Mail usages. Presumably, Facebook is more often used at an unconscious level, and is so deeply and naturally integrated in subjects' phone interaction that they did not think of the questionnaire right afterwards. An indicator for this hypothesis is that particularly in *Interval*, where only one reminder a day was sent, so many Facebook reports were forgotten.

Reliability: App Usage Duration

In all conditions, Facebook has been used more than twice as long as Mail. Comparing the self-reported usage durations, subjects overestimated their usage in all conditions by between 92% and 217%. Figure 7.3 visualizes logged (dark blue, dark green) and reported (light blue, light green) Facebook and Mail usage durations. For logged and reported Facebook usages, we found significant differences in *Voluntary* and *Interval* ($p < 0.05$) with a Student's t-test. The differences between logged and reported Mail usages were significant in all conditions ($p < 0.005$).

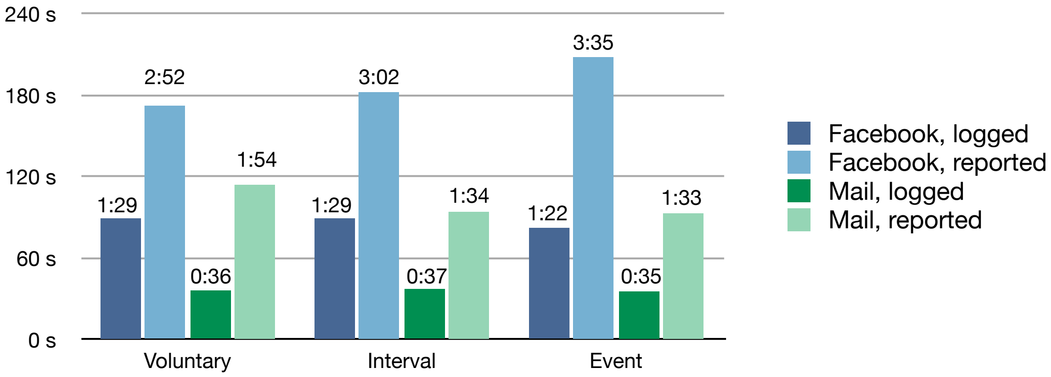


Figure 7.3: Self-reported and logged usage times of Facebook and Mail in the different conditions. Participants overestimated the durations of their actual app usage sessions often by more than 100%.

We hypothesized that subjects would overestimate their actual app usage when reporting on their behavior, which was already suggested by previous findings [72, 119]. In fact, subjects overestimated app usage durations mostly by more than 100%.

Reliability: Self-reporting over Time

Figure 7.4 illustrates the self-reporting behavior over the course of the study. The diagrams illustrate the direct self-report ratios (blue, squared graph) and total (direct plus indirect) self-report ratios (green, circular graph), aggregated for each week.

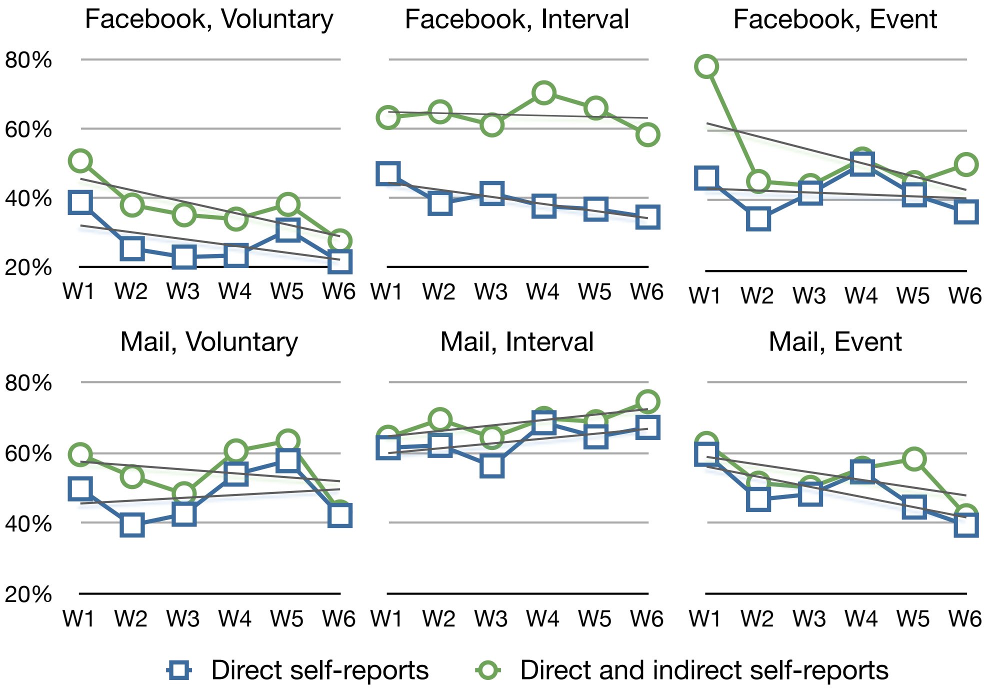


Figure 7.4: Amounts of self-reported application usage for each week in the conditions *Voluntary*, *Interval* and *Event*. Direct self-reports represent the number of filled out questionnaires, while direct and indirect self-reports also comprise app usages that have been caught up in a subsequent questionnaire. Trend lines are shown in light gray.

We can summarize the following observations:

- We observe a decrease of the self-reporting ratio in all conditions (except for Mail in the *Interval* condition) between Week 1 to Week 6 (however not equally fast), illustrated by the trend lines in Figure 7.4.
- We find a down-up-down pattern as follows: After a decrease by averagely 9.4% in Week 2, ratios remain constant and slightly rise again in the middle of the study (+6.1% from Week 3 to Week 4), but decrease by averagely 7.7% from Week 4 to Week 6.
- As already outlined in Figure 7.2, reporting rates start and remain lowest in the *Voluntary* condition. In *Interval* and *Event*, they start at a comparable level, but decrease more in the *Event* condition than in *Interval*.

Actual Behavior: App Usage Over Time

After we have looked at the accuracy of self-reported app usage (ratio of reported to real app usage), let us look at the absolute number of Facebook and Mail usages determined by logging, which are visualized in Figure 7.5. We can see that real app usages significantly decreased in all conditions, partly by more than 50% between Week 1 and Week 2. This trend partly continues until Week 4. Towards the end of the study, app usage rises again in all conditions. Still, in Week 6 there were in total 23% less logged Facebook and Mail usages than in Week 1.

		Overall	W1		W2		W3		W4		W5		W6
Volun-tary	Fb	**1190**	241	-1.3 %	238	-32.2 %	180	21.1 %	228	-88.4 %	121	33.5 %	182
	Mail	**1018**	227	-83.1 %	124	20.0 %	155	-2.0 %	152	5.6 %	161	19.1 %	199
Inter-val	Fb	**934**	234	-55.0 %	151	-19.8 %	126	-0.8 %	125	15.0 %	147	2.6 %	151
	Mail	**1184**	236	-7.8 %	219	-10.1 %	199	-30.9 %	152	18.3 %	186	3.1 %	192
Event	Fb	**1057**	237	-19.7 %	198	-41.4 %	140	-2.2 %	137	4.9 %	144	28.4 %	201
	Mail	**1429**	324	-50.0 %	216	16.0 %	257	-31.1 %	196	8.0 %	213	4.5 %	223

Figure 7.5: Logged application usage (number of Facebook, in the table abbreviated as Fb, and Mail sessions) for each week in the conditions *Voluntary*, *Interval* and *Event*. Decreases between weeks are marked red, increases are marked green.

We had asked subjects every week how reliable their self-reports were, according to their self-estimation. In fact, subjects stated that their behavior was influenced by self-reporting. The higher the "intensity level" was (i.e., from *Voluntary* over *Interval* to *Event*), the stronger subjects agreed that they changed their behavior. Overall, in the first two weeks, 0% and 3% agreed that they have not regularly filled out questionnaires. By week 3, the amount rose to 30%, by week 4 to 47%, and in week 5, more than half of participants (53%) admitted to not regularly fill out questionnaires any more. In the last week, the estimation was slightly lower again; 43% of subjects stated to have missed questionnaires.

Post-Study Feedback

In order to understand this effect, let us look at quantitative and qualitative user feedback we collected after the study. Subjects expressed their agreement to the statement *"Answering the questionnaire was low effort"* on a 5-point Likert scale (1 = strongly disagree, 5 = strongly agree). Average responses were 4.1 in *Voluntary* (both for Facebook and Mail), 3.8 in *Interval* (both for Facebook and Mail), and in *Event* 3.7 (Facebook) and 3.4 (Mail). Hence, self-reporting was less burdening in *Voluntary* than in *Interval* and *Event*.

Furthermore, participants were asked for feedback, and whether or how the study had changed their application usage. We present in the following some answers grouped by conditions[89].

Voluntary Subjects mostly liked the way of voluntary self-reporting. P3 found it *"fast and playful"* and liked that it is *"low effort and can be filled out any time"*. P6 said that *"it's simple and [...] actually uncomplicated. There's not much interface necessary."* Some subjects would have preferred some kind of automation. P5 said: *"I'd have preferred to be asked automatically, once or several times per day, to fill out a survey. Because I was in a hurry or I forgot it fairly often, I didn't fill out the questionnaire each time. A daily notification would have been sufficient for me."* For five subjects, usage habits did not change. However, five reported to use apps shorter or less frequently. P1 stated to *"use apps more consciously, only when I really wanted to use them and not just started them because there was nothing to do"*. According to P10, the effect was *"no endless surfing any more and reduced usage"*.

Interval Similar to *Voluntary*, most participants stated to like this way of reporting. One subject (P15) even would have favored *Event*, stating that *"questionnaires should pop up by*

[89] The statements were translated from German to English.

default after the app". Some participants found the effect of self-reporting interesting. *"[I was] more aware when estimating my usage time"* (P19). P13 said that it was *"interesting to yourself how often you open the apps"* and found the effort *"acceptable"*. P17 admitted to have become sloppy with the time, but did not perceive questionnaires annoying. However, six out of ten participants in this group stated to have changed their behavior. For example, they checked their emails less frequently (P13) and used the apps generally less so that they did not have to answer questionnaires (P14, P20). P12 stated to *"not have looked at every single mail, and moved Facebook usage to the PC"*.

Event Comments from participants in this condition were more critical than in the other conditions. P23 said: *"Too time-intensive and complicated. Periodically answering the same questions over and over again is annoying. Sometimes, you even avoid using those apps."* Another user suggested to use less clicks in the questionnaire to make self-reporting more convenient. P26 did not like event-based reporting because it was *"too annoying"* and he would *"prefer background logging"*. Nine out of ten subjects also indicated that they used Facebook and Mail differently or at least thought about it. P21 said: *"I partly looked up e-mails at my PC when I was too lazy to fill out the questionnaire on the mobile"*. Other participants reported to have used the apps significantly less, especially towards the end of the study (P22, P24, P25). P23 stated to *"often have read only the notification but not started the app any more"*. Unlike most other participants in the *Event* condition, P28 was happy about the increased awareness of app usage: *"I now know how much time I've wasted with that! I should waste my time with other things."*

Two weeks after the study had ended, we contacted the subjects again. We asked how long they would be willing to participate in a similar study. 22 of the 30 initial subjects answered this question, which is depicted as summary in Figure 7.6. The median length named by subjects is four weeks. It becomes evident that the answers differ based on the conditions in which subjects had participated in the study. The *Event* participants suggested intervals between zero and three weeks, the *Interval* participants intervals between one and six weeks, and only the subjects from the *Voluntary* condition could imagine longer study durations, with a length up to twelve weeks.

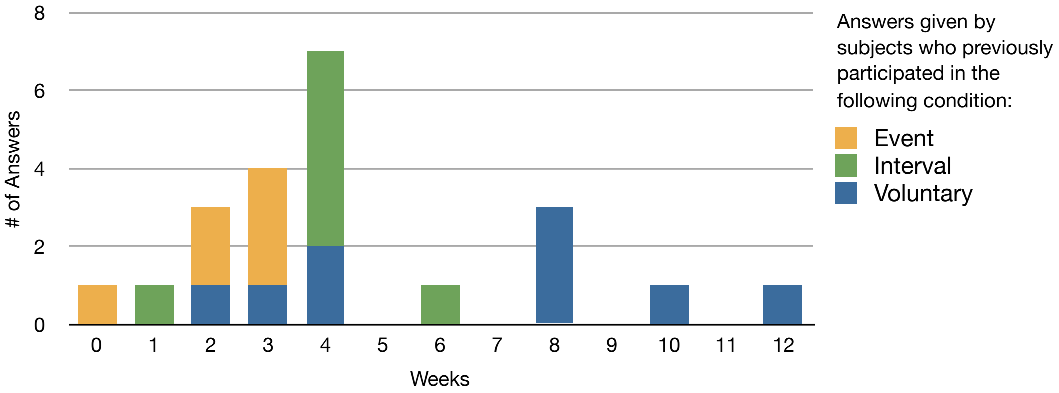


Figure 7.6: Histogram of the maximal acceptable duration of a study in which participants would participate again in a study with the same conditions (answers of participants in respective conditions are highlighted in different colors).

7.4.3 Discussion and Lessons Learned

After we have presented the results, we try to answer the initially formulated research questions.

Reliability of Self-Reports (RQ1–R3)

RQ1–RQ3 addressed the reliability of self-reports. We could identify the following findings.

- In terms of reporting accuracy, subjects overestimated their application usage durations in all conditions.
- In terms of the quantity of reports, only approximately 40% to 70% of actual app usages were reported by subjects within six weeks. The intensity of reminders did not have a significant effect on these ratios.
- Over time, self-reports decreased by averagely 23% within six weeks.

These results show that self-reports on application usage can generally not be considered as accurate. These findings show that we need to be aware that self-reported data can be unreliable. Researchers should not follow the pitfall of blindly trusting self-reported data.

Influence on Actual Behavior (RQ4)

Regarding RQ4, we found that self-reports actually can alter the behavior of subjects. While ups and downs in app usage are normal (e.g., due to holidays), it is likely that subjects altered their application usage because of the study. This behavior is known as *Hawthorne effect* [284]. Both quantitative and qualitative results indicate that subjects reduced their actual application usage in order to avoid questionnaires. This shows that a study setup needs to be carefully planned in order to really measure the intended effect (i.e., the "unobserved" state). With the answer to the next research question, we try to give recommendations to avoid this problem by optimizing the duration of a study and the self-reporting frequency.

Ideal Self-Reporting Frequency and Study Duration (RQ5)

With RQ5, we aimed to investigate the ideal study duration and frequency of self-reports. This question cannot be answered definitely. From a researcher's perspective, a higher self-reporting frequency would be desirable, as it generates more samples. Subjects answered more than 90% of all questionnaires that automatically popped up in the *Event* condition. However, in a long-term study lasting significantly longer than six weeks, this frequency of automated questionnaires would most likely not be feasible, since it would annoy users too much. Even in our setup in the *Event* condition, where only for 49% of app usages a questionnaire appeared, subjects found reporting too time-intensive and annoying. The differences between conditions are smaller as anticipated, which is in line with subjects' feedback that self-reporting felt burdensome already in *Interval* and *Voluntary*. We also saw that a higher "demanded intensity" does not correlate with a higher amount of reported instances; thus, the intensity should reasonably be limited.

While some participants in *Interval* wished for automated surveys to prevent that they forget to report (which is exactly what we provided in the *Event* condition), subjects in *Event* felt overly burdened and wished for a larger interval between reports, or for logging. This shows us that reasons for inaccurate self-reporting are twofold: First, subjects may forget about reporting, which was the case in the *Voluntary* condition. Second, subjects may deliberately not answer questionnaires because they find it too much effort; this was the case in the *Event* condition. Some recommendations could therefore be:

- Do not use an overly high self-reporting intensity or interval. In our study, report rates initially were below 70%, with further decreases from the second week on. Thus, dense self-reports from participants are hard to justify. If the burden is too high, participants will get annoyed so that they refrain from reporting, or more severely, the alter their actual behavior. Given that reporting rates in *Interval* were in average higher than in the (more demanding) *Event* condition (although not significantly), less "pressure" can probably lead to more satisfying results at the end of the day.

- Based on the development of self-reporting rates over time, we learn that it is challenging to maintain commitment over a duration of six weeks. The optimal study length depends on the required reporting intensity. For example, Facebook usage reporting in the *Event* condition already significantly dropped after Week 1 (so that one to two weeks would be ideal). By contrast, a study with *Interval*-based reporting could last four to five weeks. Based on questionnaire answers, the mean for a maximal study duration would be four weeks (see Figure 7.6). However, the great differences between answers of participants who had experienced different conditions support the assumption that higher intensities, such as in the *Event* group, reduce the reasonable study length.

- Researchers should be aware that self-reports are not accurate, and consequently take into account that a corrective factor might be necessary when analyzing and interpreting the results. Given that the highest drop in self-reports occurred right at the beginning of the study, researchers could consider to really start data collection after some days of "test run". This gives subjects time to habituate with the reporting procedure. In order to find out how subjects can be kept engaged from the beginning, this question will have to be further investigated in future experiments.

- While the reporting rate decreased in total over the course of six weeks, there was a relative increase in the last third of the study. This increase might be due to reminder emails we sent out to participants. The reminders may have strengthened the sense of duty of those who had neglected reporting towards the end. Subsequent work could systematically investigate the effect of reminders (regarding, e.g., number or frequency, since too frequent reminders could have a contrary effect).

Finally, as we compared self-reporting to logging, we would like to point out directions to decide for either of the two data collection methods.

- If quantitative data shall be collected, logging is often ideal to capture data in a reliable and unobtrusive way. Self-reporting may be an option for situations where automated data capturing is not possible.

- Diary reports might not reflect entirely reliable usage frequencies, but this does not make them less reliable for the assessment of the experiences recorded. The qualitative nature of self-reports has the unique advantage to gain insights that would not be possible with automated data collection, such as motives and decisions of users for the behavior that was logged.

- We showed that self-reporting can even influence actual behavior, in our case the usage frequency of the observed apps (and it is critical when the research method influences the observation!). If possible, researchers should therefore consider a combination of self-reports and logged data to achieve additional certainty, or to use a control group in the experimental design. As of today, where self-reports are often collected with the help of smartphones, automated collection and logging of information is in many cases a small extra effort. The SERENA framework presented in this chapter can support researchers to this end, as it provides logging and different ways of self-reporting (interval and event-based).
- The data collection method should thus carefully be adapted to the scenario and to the actual data to be gathered. If, e.g., just random experiences or impressions should be collected, it can even be preferential when users do not report too often. In that case, reporting occurrences become indicators for moments that are salient or meaningful to subjects. However, when quantitative data or "instances" should be captured, self-logging can provide unreliable and incomplete data.

7.5 Research in the Large

After we have in the preceding section discussed long-term evaluation and thus tackled the time dimension of user studies, we now look at best practices for large-scale results in terms of participants. We focus on the two approaches crowdsourcing and marketplace research, which we identified as particularly useful in the context of MUSED systems.

The term "research in the large" has been coined by a workshop series started in 2010, which was held in cooperation with the UbiComp and MobileHCI conferences [64, 65]. The goal was, in the first place, to investigate strategies for collecting quantitative data from a large user basis for statistical analysis [28] in the context of Ubiquitous Computing (UbiComp) research. We extend on this notion by including also large-scale *qualitative* data, which can for example be gained through crowdsourcing intelligence.

7.5.1 Crowdsourcing Human Intelligence

In the context of UbiComp, the approach of acquiring information or data from "the crowd" has been used especially in the sense of crowd*sensing*, i.e., using data from a large number of individual, sensor-equipped devices [108]. By contrast, crowd*sourcing* is a means to also gain large-scale (especially qualitative) user feedback.

Amazon Mechanical Turk, or briefly mTurk[63] has been launched in 2005 and is a so-called micro-task market for coordinating HITs (Human Intelligence Tasks). Requesters can place tasks that are accomplished by providers for a compensation. That way, a large user basis can be recruited, e.g., for online surveys or questionnaires. Various researchers have attested mTurk to be a viable alternative to traditional subject recruitment methods [37, 156, 258]. Particular advantages of mTurk are the demographic spread and the geographic distribution of users, resulting in a higher diversity of participants than in many studies with the typical "American college" population [37]. This can especially be advantageous if results shall be representative for wide parts of the population with cross-cultural background. On the other hand, limitations to a subset of users are possible, e.g., in order to investigate effects in particular cultural circles or target groups. Data gained through portals like mTurk are as reliable as obtained with traditional methods [37], but Kittur *et al.* [156] point out the importance of the survey design. For example, they recommend to use verifiable questions and to design tasks in a way that producing malicious responses does not take significantly less time than valid ones. There exist numerous alternative websites like ClickWorker[90], Crowdtap[91], ShortTask[92], or MobileWorks[77], offering similar services.

In our research, we have used crowdsourcing especially in early stages of the design process. In Chapter 3.2, we investigated people's attitudes towards different object interaction techniques in the context of MobiMed, a medication package identifier. With mTurk, we gained valuable large-scale insights on what people perceive as advantages and disadvantages. The method gave us more diverse feedback than we could possibly have gained through personal interviews or surveys in the closer entourage of the researchers.

In Chapter 5, we used MobileWorks to evaluate a mockup of an indoor navigation system comparing different instruction types. The crowd-sourced feedback helped us to confine ourselves to the designs which were evaluated best in the online survey, and to improve the initial concept in a subsequent prototypic implementation.

7.5.2 Marketplaces as Data Source for Large-Scale Usage

As next aspect in the context of research in the large, we discuss the usage and meaning of digital marketplaces for research. We use the term "marketplace" for a central place where apps are distributed and installed for a mobile platform, in order not to confuse terms with platform-specific stores (e.g., the App Store on iOS, or Google Play on Android). Böhmer and Krüger [26] suggest the term "research in the application store" when marketplaces are used, as the general term "research in the large" could also include other distribution sources of research apps (e.g., email).

We discuss three aspects of marketplaces in this section. First, we give some examples for marketplace research from prior work. In particular, we illustrate marketplaces as a chance to obtain large-scale data. Second, we discuss advantages and problems of marketplaces as a means for deploying research apps. Third, in a case study, we investigate update behavior with marketplaces, which may not only be security-critical, but also problematic in the context of research app deployment.

Large-Scale Studies in Marketplaces

Research applications have been deployed via marketplaces in several research projects [25, 99, 123, 124, 209, 292, 293]. Böhmer *et al.* [25] analyzed application usage on mobile devices. They investigated (for different categories of apps) application launches, sessions, transitions, and time-dependent usage. As vehicle to monitor these data, they used an app

[90] `http://www.clickworker.com`, accessed March 11, 2014
[91] `http://home.crowdtap.com`, accessed March 11, 2014
[92] `http://www.shorttask.com`, accessed March 11, 2014

recommender tool[93] that suggested interesting apps for download based on usage patterns. Ferreira *et al.* [99] analyzed smartphone charging habits with the help of an application called SuperCharged that helped users to monitor their battery life.

Games turned out to serve particularly well as vehicles for research applications, as Henze points out: “Players execute the strangest tasks.”[94]. McMillan *et al.* [209] report on app usage in the wild determined based on a game deployed via the iOS App Store. Kranz *et al.* [171] inquired into the adoption of the NFC interaction technique by analyzing the usage of a NFC-based game. Henze *et al.* [123] investigated touch performance on mobile devices with different target sizes and screen locations. Data from over 90,000 subjects was gained through an engaging game called Hit It![95], where randomly appearing circles had to be touched as fast as possible. A similar study by Henze *et al.* was conducted on typing performance [124] using a speedwriting game[96].

Sahami *et al.* [293] investigated preferences for mobile notifications. They deployed an app that mirrors incoming smartphone notifications on a computer (similar to Apple’s “Continuity” feature[97]). Over 40,000 users downloaded the app[98] in Google’s Play Store and thereby generated data for the study. The same researchers also investigated smartphone postures for different tasks with the help of a widget deployed in the Google Play Store [292].

Besides the explicit intention to distribute research apps, marketplaces have been used for various purposes in a research context: For example, download numbers can serve as an indirect indicator for acceptance and proof of concept [336]. Zhai *et al.* [350] gained qualitative feedback on their application from user comments in the app store. Miluzzo *et al.* [215] looked at implications and challenges of large-scale distribution of research apps. They pointed out that insufficient software robustness and poor usability may lead to a loss of confidence on the part of the users. However, they did not quantitatively examine this phenomenon (such as the number of uninstalls due to dissatisfaction). For further experiences with deployment of research apps in marketplaces, we refer the reader to Boll *et al.* [28], who summarize their experiences based on different research projects. They particularly discuss supplemental aspects like validity and ethical considerations.

Advantages and Disadvantages of Research App Deployment in Marketplaces

Let us now summarize the most significant advantages and disadvantages of marketplaces and the consequences for research app deployment.

[93] appazaar, not available for download any more

[94] http://de.slideshare.net/NielsHenze/how-to-do-mobilehci-research-in-the-large, accessed July 2, 2014

[95] https://play.google.com/store/apps/details?id=net.nhenze.game.button2, accessed March 10, 2014

[96] https://play.google.com/store/apps/details?id=net.nhenze.game.typeit, accessed March 10, 2014

[97] https://www.apple.com/ios/whats-new/continuity/, accessed September 17, 2014

[98] https://play.google.com/store/apps/details?id=org.hcilab.projects.notification, accessed March 10, 2014

Advantages

With marketplaces, a high user base can be reached. Potential participants of a study are thereby likely to be more heterogeneous regarding their background, country of origin, etc., compared to "on-site" recruitment.

Marketplaces have furthermore the advantage that they are actively searched by smartphone owners. If the research app provides a concrete benefit beyond the purpose of collecting research data, offering the app in the marketplace can be a good strategy to acquire a large number of participants for large-scale research. Users are attracted by the functionality and, as a side effect, provide their usage data for research purposes. However, the app description should explain that the application is part of a research project and the collected data should be made transparent.

Another advantage is the ease of the installation process. Most users will be familiar with the process of finding and downloading apps from the marketplace. For research apps, it will thus be easier to make users install the app via the marketplace, instead of using a custom deployment procedure. For non-marketplace deployment, the executable file would either have to be sent via email (which might be problematic for larger applications), to be downloaded from a website using the mobile device (which requires multiple steps of interaction), or to be transferred using a wired connection from a computer. All these methods would require more technical support than a marketplace installation, and would probably even require participants to come to the research institution to get their device prepared for the study.

Moreover, in the default configuration of Android, installation sources other than Google Play are disabled. When research apps are directly installed without marketplace, subjects not only have to change this setting (which requires additional installation steps), but also the security warning that appears when allowing any app for installation might make subjects concerned that participating in the study might do harm to their device (in fact, there *is* an increased security risk in that case). On iOS, there are anyway limited options to deploy applications other than using the App Store. Developers can use ad-hoc deployment, but the device ID must be known and added to a provisioning profile. Again, this requires the additional step that potential study participants must look up and send their device ID beforehand to the researchers. Furthermore, this deployment method is restricted to 100 devices by Apple, which might be too small for some large-scale setups. The second alternative is to use the enterprise deployment option. However, this deployment method is a legal gray area, as it is only intended to be used within a company.

Disadvantages

There are, on the other hand, several drawbacks of marketplace deployment.

First, marketplaces make applications public. This proceeding is referred to as "quasi-experimental design" [305], i.e., the researchers are not able to assign or choose participants as easy as with other recruitment methods. Depending on the study setup, it may not be desired that everyone has access to a research app, e.g., because a special target group shall be observed, or because the developers want to use a certain recruitment process (e.g., to be in control of demographic parameters).

Another reason why public access to research applications might be unwanted is the potential disclosure of the researching institution, which contradicts the “blind review” policies of many publishers. To overcome this drawback, apps could be marked as alpha or beta versions (which is possible in Google Play), which restricts the download to registered testers. Another possibility would be to require in-app activation, so that only test subjects who have received an activation code can use the app. Similarly, a (deterrently high) price in the store could be used to drive off “normal” users, while test subjects could receive a promo code they can redeem to download the app. However, all these solutions require a separate channel for distributing those codes, which again complicates the setup.

A further point to be considered is the (one-time or annual) developer fee that is a prerequisite for offering apps via a marketplace. This is probably not a decisive factor for research institutions, but still should be listed for completeness.

Moreover, marketplaces have different levels of checks before an application is accepted. Apple uses a set of guidelines to review apps before they are accepted in the store[99]. Although Google Play is free of constraints for uploading apps, they are filtered for unwanted behavior [192] and, in case of malicious content, deleted. This policy is appropriate to keep up a general level of security; however, it might filter out research applications that alter the operating system behavior on a low level (which might in some cases be necessary for research purposes).

Finally, it might be required for research applications that the experimenters can react in time on problems and provide updates. While app stores provide an integrated update management, there are still different issues to consider when it comes to update behavior, which we discuss in detail in the subsequent section.

Update Behavior in Digital Marketplaces – A Case Study

A central element of digital marketplaces is the ability to receive updates of installed applications in one place. The various mobile platforms implement the update mechanism differently (we here focus on Android and iOS as most prevalent platforms). In Apple’s iOS, up to version 6, users were informed about available update via a badge symbol on the App store icon, but updates were not installed automatically. Beginning from iOS 7, users can choose to download and install updates automatically (but not make this setting on a per-app basis). Android provides automatic updates since version 2.2; newer Android versions allow to configure if all apps are updated automatically, or individual apps shall be excluded. If automatic downloads are disabled, users are notified on updates via a message in the notification bar. For research applications, the timely installation of updates is important for several reasons:

- Updates can change or update functionality, e.g., change the study condition in a within-subjects experiment. All participants should therefore install the update at the same time.
- Updates can add missing features or fix bugs. Failure of installing the update could distort recorded data, or even prevent that data is collected at all.

[99] https://developer.apple.com/appstore/guidelines.html, accessed March 10, 2014

- Apps could contain security holes, which increase the vulnerability of the device. In particular, research apps under active development might be less stable than established commercial applications, and therefore require more frequent fixes.

We report on a case study where we observed users' update behavior and gained insights on the correlation between published updates and their actual installation, and discuss the consequences. For this, we used the application VMI Mensa[74], developed and maintained in our research group. It shows meals and prices of Munich cafeterias and canteens and has become very popular with students since its launch in July 2011, with more than 5,400 downloads in July 2014. It has received 200 ratings (averagely rated with 4.7 out of 5 stars) and 67 user comments.

Method

We looked at five consecutive updates of VMI Mensa between December 2011 and April 2012. For our analysis, we used the built-in statistics tools of the Android Developer Console[100] in Google Play. Using this tool, we were able to keep track of the number of installations over time, and the distribution of currently installed app versions. The data is anonymous and cannot be related with individual users.

The average time between updates was 26 days, which we do not consider as unreasonable effort for users to regularly install them. All updates added new functionality to the app and/or fixed small problems, but none were critical for security. The "Recent Changes" description in Google Play was adapted for each update to make transparent what the benefits of the updates were. For each update, we observed how many users downloaded it on the initial day of publishing and in the 6 consecutive days. We calculated the update installation ratio by relating the download count to the total count of active device installations on the respective days.

In addition to the anonymous update installation statistics, we considered available user communication in form of feedback emails (provided in the Google Play description), comments, and ratings in Google Play for our analysis. We will bring in these findings in the discussion section.

Results

Figure 7.7 illustrates the number of update downloads over time. We can summarize that most users who actually *do* install updates install them in the first few days. Those who do not install them early are also not likely to do so in the subsequent days.

Averaging over all five updates, we found that 17.0% installed the update on the day it was published (day 0). On the following days, the numbers continuously and exponentially decreased: 14.6% installed the update on day 1, only 7.8% on day 2, and 5.1% on day 3. On day 6, only another 2.3% downloaded the update. In total, slightly more than half of all users (53.2%) had installed the updates after one week. The standard deviations of these percentage values between the five updates we considered lie between 0.4% and 2.7%, indicating that the update behavior was very similar for all five updates.

[100] https://play.google.com/apps/publish, accessed March 10, 2014

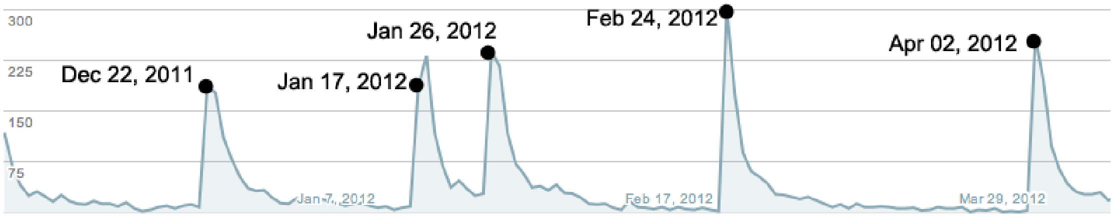


Figure 7.7: Download number of five subsequent updates (vertical axis) over time. The graph shows maxima on the update publishing day (possibly also due to activated auto-updates) and exponentially decreases thereafter. Figure based on Android Developer Console graph.

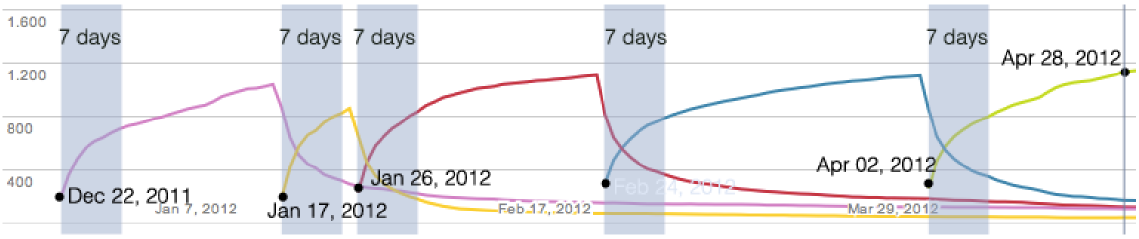


Figure 7.8: Installation number by version (vertical axis); the colored lines indicate the five latest versions. The diagram reveals how long old versions are active on user's devices. The 7-day periods after an update has been published are highlighted. Figure based on Android Developer Console graph.

Looking at the fractions of the latest five versions (illustrated by different colors in Figure 7.8), we see the slow increase of new versions due to cumulative installs (visualized with a steep graph that flattens out more and more), and the decrease of older versions. The seven-day periods after an update has been published are slightly shaded for illustration.

It becomes evident how long outdated versions (up to four versions older than the latest one) are still circulating. As an example, let us look at April 28, 2012, which is two weeks after the latest update has been published: Only 56.4% of all users have installed the latest available version at this time. The previous four versions were still in use by 8.5% (v.26), 6.0% (v.25), 5.5% (v.24) and 2.1% (v.23). Most severely, 21.5% had even older versions installed on their devices at that time.

Discussion

With only half of all users installing updates within one week, and one fifth not even installing one of the last five updates, developers must assume that a significant amount of users runs outdated app versions. We hypothesize that the relatively high ratios of the first two days might partly be due to an amount of users who enabled the automatic update option. There might be several reasons why not all users use automatic updates:

- The function is disabled by default (this was the case at the time when the following study was conducted).
- Users disable updates manually to save network or processing resources (especially older devices tend to slow down with an update process running in the background).

- Users are afraid of accidental mobile data usage.
- Users want to be “in control” about installations they make (updates sometimes do not bring improvements, but sometimes unwanted functionality like advertising or aggressive privacy changes, so that the prior app version is preferable).

The findings of this case study add an additional perspective to the pro and contra arguments to distribute research apps in marketplaces. Relying on the built-in update functionality does not guarantee timely updates, which can be important for research applications. One clear recommendation is therefore that researchers should try to keep the reliance on updates as small as possible. For example, a change of study conditions in a within-subjects study should not be realized by an app update. Instead, such actions should be triggered through a connection to a server, or already foreseen at development time and hard-coded in the app itself. Another solution could be to provide an own update mechanism built into the application, which either automatically downloads updates in the background, or notifies users on an available update within the app, and forwards them to the marketplace where they can download it.

It must also be considered that our case study only looked at one app, independently of the total number of apps installed. A high number of installed apps could further decrease the amount of up-to-date apps, since more time would be required to keep all apps up to date. A further limitation of this case study is that the target group for VMI Mensa are students. A different behavior could be shown by other target groups. However, we hypothesize that students as potentially above-averagely technology affine users update more frequently than average users. This would entail that general update rates might be even lower than observed in this case study.

While this section focuses on the impact in research, let us make a final note on the “lazy update problem” in other contexts. With VMI Mensa, we chose a frequently-used app as vehicle for our case study. However, the motivation for keeping occasionally used applications up to date might be even worse. This is problematic, e.g., for online banking, where security problems are particularly critical. In order to better understand the relation between usage frequency and update behavior, in-depth usage monitoring [25] is required. To improve the overall security level on a device, special concepts like compartmentalization in different security zones have been proposed [276].

7.6 Summary: A Suggested Evaluation Process for MUSED Systems

Based on our experiences, we suggest the following proceeding as exemplary model for the iterative evaluation of multimodal and sensor-driven (MUSED) systems. For a summary of goals and important attributes of each involved research method, see Table 7.1. Prior to this proposed evaluation process, we presume a thorough review of related work (thus, literature review is not mentioned as separate step). Depending on the available prior knowledge (gained from own prior research, related work, etc.), individual steps of this process may be omitted in practice. To inform this decision, we now discuss the most important characteristics of each step.

Method	**Focus of Interest**	**Why Important?**
Focus Group	Initial insights on user needs	• Fast • Inspiring • Directions for research questions
Large-Scale Questionnaire	Broad feedback on concept or mockup	• Early feedback prior to implementation • Heterogeneous participants
Laboratory Evaluation: Wizard of Oz	Quantitative and qualitative hands-on feedback	• User experience with prototype • Controlled conditions for measurements • Face-to-face interaction with participants
Real-World Evaluation	Usage and adoption in the field	• Relation to context and environment • Degree of novelty • Long-term usage

Table 7.1: Proposed iterative design model for MUSED systems based on our research experiences.

Step 1: Focus Group

The goal of this method is an initial discussion of ideas and concepts with a small number of people (see Section 2.4.1). In the context of MUSED systems, focus groups can reveal users' attitudes towards novel interaction methods and give insights on their wishes and needs. That way, a focus group discussion can give directions for research questions to be investigated and inspire concepts and designs. We started our investigations of multimodal rule creation and awareness interfaces (see Chapter 6.4.1) with a focus group discussion, which helped inform the later designs.

Step 2: Large-Scale Questionnaire

After initial concepts and research questions have been framed, large-scale questionnaires (conducted as online survey) help obtain broad feedback. They can be used for different purposes: forming an image of the status quo, estimating attitudes towards a new concept, or getting concrete feedback on a presented interaction technique or system. A particular advantage of the large scale is that participants can be very heterogeneous depending on the recruitment method. Subjects of varying age, with different backgrounds, occupations, and experience can bring up issues researchers probably did not think of before. Crowdsourcing, e.g., using Amazon mTurk, is one valid method for obtaining large-scale questionnaire results. Based on initial results, it is often possible to identify concepts that are worth deeper investigation out of several alternatives, and exclude those who were rejected by participants. In our research, large-scale surveys successfully provided detailed information on multimodality usage (Section 6.2.1) as well as feedback on physical interaction modalities (Section 3.2.3) and indoor navigation visualization (Section 5.3.2).

Step 3: Laboratory Evaluation: Wizard of Oz

The next step should, at some point, be a laboratory experiment with a prototype. Many quantitative measurements, including efficiency, effectiveness, and usability, can only be taken in a hands-on study. These measurements can serve to compare various design alternatives determined in *Step 2*, or to compare a novel technique with an established "baseline" technique. However, laboratory evaluations can also provide valuable qualitative findings by observing users or by employing the "think aloud" method [328]. As motivated in Section 7.2, we found the WOz technique particularly suitable for multimodal systems, why we recommend using this approach in a laboratory study. This dissertation has presented a variety of research where lab studies in conjunction with WOz were employed. See therefore, e.g., Sections 3.2, 5.3.3, 5.4.2, 6.4.2, and 6.5.2.

Step 4: Real-World Evaluation

A real-world evaluation in the field can be the final puzzle piece in the evaluation of a MUSED system. As multimodal interaction approaches often involve contextual factors, some research questions can only be answered *in situ*. With logging and self-reporting, we have presented two applicable methods for real-world data collection. While logging through research apps can give detailed usage information (but not qualitative insights), self-reporting can provide the missing qualitative perspective (e.g., users' motivations, goals, and thoughts). However, self-reports can be cumbersome in the long run, as discussed in Section 7.4.3. Studies over a longer period of time can for example reveal if users adopt a technique, how quickly they familiarize with it, or if they use it under certain conditions (e.g., when they are in a hurry). As distribution channels, application marketplaces can be used to reach a heterogeneous group of users and to conduct "research in the large". Advantages and pitfalls of this approach have been discussed in Section 7.5.2 in this chapter.

Part IV

Conclusion

Chapter 8

Conclusion and Outlook

8.1 Summary of the Contributions

At the end of the dissertation, we give a summary of our main contributions in the individual chapters. We also refer back to the (high-level) research questions (HRQs) formulated in the introduction in Section 1.2.

In Chapter 2, we situated our work in the body of related research. By outlining the state of the art in different application domains, we already showed the potential for improvements through multimodal and sensor-driven interaction, and thereby partly answered **HRQ1**. We also coined the term of MUSED interaction for our focus of interest in this work, and recommend to use this term whenever the sensor-driven aspect of multimodality shall be emphasized.

Chapters 3–5 gave a thorough view of multimodality in practice. Due to the heterogeneous requirements in different application domains, separate chapters are dedicated to three of these domains, each investigating the individual systems in depth. For each domain, we show how mobile interaction can benefit from multimodality (**HRQ1**). In each of these three chapters, we also addressed potential challenges, addressing **HRQ2**. In Chapter 3, we investigated how MUSED interaction supports users in daily health-related tasks. Therefore, we presented natural modalities for physical object interaction (touching, pointing, and scanning), which were evaluated to be faster than traditional, text-based mobile interaction. We further investigated how MUSED interaction can enable novel interaction in the context of personal fitness. To this end, we developed with GymSkill a skill assessment system for physical exercising. We argue (supported by questionnaire results) that by individualized feedback on how to improve in an exercise, long-term motivation is fostered. The specialty of this approach is that the (multimodal) interaction with GymSkill is contained in the training procedure with the fitness device. Chapter 4 is dedicated to education. We here investigated how university employees and students can benefit from MUSED interfaces in teaching and learning environments. We proposed MobiDics, a context-based didactics support tool for docents; and depicted a holistic scenario for students and lecturers with several systems that are the result of our research: MobiDics, Ubiversity, and an interactive door sign. The presented lineup of systems showed a variety of different interaction modalities and classes (e.g., public-private interaction). The question how the challenges of visual indoor localization can be addressed by a MUSED user interface is treated in Chapter 5. We here proposed a VR-based visualization which was evaluated to be superior to AR-only in navigation time and perceived accuracy. We also made suggestions for a combination of both approaches to use the strengths of each of them for an improved overall navigation experience. We further introduced special UI elements (feature indicator, and object highlighting visualizations), and showed how they improve both the user experience and add to increased system accuracy. The heterogeneity of employed input and output modalities (summarized in Figure 1.3) of the presented systems illustrate the breadth of a design space for multimodality. By providing concrete solutions for each of the chosen exemplary domains, we answered **HRQ3**.

In the second part of the dissertation, we took a more abstract point of view, and focused on the development process of MUSED systems, informed by the lessons learned from the in-depth studies in Chapters 3–5.

Chapter 6 investigated two major aspects. First, how can the implementation of multimodal systems be simplified? Second, how can users be supported in adapting the multimodal behavior of mobile systems to their needs, and how can awareness on currently active modalities be achieved? Relating to the first question, we proposed an extensive software framework (M3I) to leverage the creation of multimodal applications for developers, and presented examples that demonstrate the feature space and capability of M3I for rapid prototyping of multimodal interaction (answering **HRQ5**). Addressing the second question, we decided for a rule-based system as solution for easy-to configure, system-wide, user-level multimodality support (informed by a focus group and literature analysis). Based on laboratory and field studies, we made recommendations how the rule creation process should be designed (e.g., situation- versus modality-oriented), and how awareness on modality changes can be increased (here, notifications turned out as best compromise between obtrusiveness and control). With this contribution, we answered **HRQ4**. Chapter 6 treated problems and challenges of MUSED interaction on a system level, and thereby also addressed **HRQ2** in a more holistic way than the application-specific chapters.

Finally, in Chapter 7, we investigated appropriate evaluation methods based on the characteristics of MUSED systems. The major contribution of this chapter is an evaluation scheme for multimodal systems, based on our research experiences. Thus, together with Chapter 6, this chapter answers **HRQ6**. We promote WOz for laboratory studies and give recommendations how to design long-term experiments in the field. We also compared the accuracy of logging and self-reporting. Further, we discussed research in the large using application marketplaces.

Our research illuminated multimodal systems in all of the three dimensions introduced in Chapter 1.2 and visualized in Figure 1.1. The *interaction* dimension was examined in Chapter 6, where both novel multimodal input methods and output modalities were proposed, discussed, and evaluated. The *abstraction* dimension manifests in the difference between the first and the second part of the dissertation. While Chapters 3–5 provide insights to specific research fields, Chapters 6 and 7 discuss the design, implementation and evaluation of multimodal systems on a general level. The *perspective* dimension is represented in Chapter 6, where we support both the developer's view with our M3I framework, as well as the end user's view with an investigation of modality awareness visualizations and rule creation approaches.

8.2 Outlook and Future Research

Besides the profound contributions of this dissertation (see Section 8.1) which have been published on renowned conferences and in distinguished journals, there are aspects we could not look at in detail in the scope of this work. These open up several possible directions for future research.

Follow-Up Research Projects

The individual projects we have presented from different application areas in Chapters 3–5 served as examples for the potential for MUSED interaction. There are, however, manifold possibilities to refine and extend these approaches, and we have outlined some of them at the end of the individual chapters. For several of these projects, research was still continued at our research institution within cooperations (e.g., in case of MobiDics), research projects, or spin-offs (e.g., in case of indoor navigation). The ideas, concepts, and scientific results of this dissertation contribute to this future work.

Adoption of M3I

Besides our recommendations and lessons learned, we have introduced the M3I framework for mobile and multimodal interaction. M3I already supports a range of context factors and triggers, but its functionality is yet to be extended, also with regard to future mobile operating system versions that provide new APIs and possibilities. We hope that the framework is adopted and further extended by the community, which is simplified by its open and modular architecture.

Novel Application Areas for Mobile Multimodality

There are several current trends of mobile interaction that are potential candidates for MUSED interaction. This includes the “Quantified Self” movement (which we have partly already addressed in Chapter 3), but also the emerging field of wearable computing, such as wearable fitness devices, smartwatches, and smartglasses. In concordance with Mark Weiser, who shaped the famous quote “The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it” [342], such devices do not use traditional forms of interaction any more. They will necessarily be accompanied by novel interaction modalities.

However, we argue that mobile devices – particularly smartphones and tablets – as we know them now, will remain the center of mobile interaction in a mid-term perspective. Smartwatches or HMDs are not intended to replace, but complement smartphones, taking over individual functions they can handle better than the phone itself. As such, smartphones serve as hub to which other “smart” devices are (wirelessly) connected. This is somewhat similar to the “digital hub” principle of desktop computers for music players, digital cameras, and other peripheral devices before the emergence of cloud services[101]. Therefore, we believe that the findings on MUSED systems presented in this thesis will remain relevant in the future, but they will have to be extended, e.g., in the direction of between-device interaction.

Other Perspectives of MUSED Interaction

This dissertation intentionally focused on an HCI perspective of multimodal interaction. Future work could investigate other perspectives, such as a computer science, machine learning, or software engineering perspective. From the low-level point of view, e.g., context detection and classification, as well as estimating the user's intention could be investigated. In terms of software engineering, the impact of multimodal interaction on organization and modeling can be researched, e.g., regarding testing, maintenance, but also (agile) project management and user-centered design, but this is beyond the focus of this work.

Social Implications

Another future research direction is the social dimension of novel interaction modalities, such as the impact of social settings on modalities. For example, not all device gestures are equally accepted in public spaces [2]. We already have dealt briefly with privacy in Chapter 6, where we discussed usage data collection as requirement for proactive modality switching, and in Chapter 5, where visual indoor navigation could be misinterpreted as video-recording other people. Future work could investigate in depth how MUSED interaction is related to privacy. With the advent of HMDs, this topic will become even more of interest for research as, e.g., wearers of Google Glass have been attacked in public[102].

8.3 Concluding Remarks

We sincerely hope that the contribution made by this dissertation will help and inspire future research in the domain of multimodal and sensor-driven interaction. We believe that MUSED interaction, as described and investigated here, will increasingly gain importance and prevalence. The insights, lessons learned, recommendations, and practical tools provided by this work shall contribute to a better user experience with MUSED systems from an HCI point of view.

The research tools presented in this work are shared with the community and offered for download. M3I is available at `http://www.eislab.net/m3i`. SERENA is available at `http://www.eislab.net/serena`. GymSkill and VMI Mensa can be downloaded from Google Play[67,74].

[101] `http://www.forbes.com/sites/briancaulfield/2011/06/03/a-decade-after-steve-jobs-introduced-the-digital-hub-icloud-will-let-apple-kill-your-pc/`, accessed July 4, 2014

[102] `http://mashable.com/2014/04/13/google-glass-wearer-attacked/`, accessed August 5, 2014

List of Figures

List of Tables

Bibliography

[1] G. Abowd. Classroom 2000: An Experiment with the Instrumentation of a Living Educational Environment. *IBM Systems Journal*, 38(4):508–530, 1999. 30

[2] D. Ahlström, K. Hasan, and P. Irani. Are You Comfortable Doing That?: Acceptance Studies of Around-Device Gestures in and for Public Settings. In *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI)*. ACM, 2014. 190

[3] A. Ahmadi, D. Rowlands, and D. James. Towards a Wearable Device for Skill Assessment and Skill Acquisition of a Tennis Player During the First Serve. *Sports Technology*, 2(3-4):129–136, 2009. 25

[4] M. Ally. Using Learning Theories to Design Instruction for Mobile Learning Devices. *Mobile Learning Anytime Everywhere – A Book of Papers from mLearn 2004*, pages 5–8, 2005. 28

[5] M. Angermann and P. Robertson. FootSLAM: Pedestrian Simultaneous Localization and Mapping Without Exteroceptive Sensors—Hitchhiking on Human Perception and Cognition. *Proceedings of the IEEE*, 100(13):1840–1848, 2012. 31

[6] D. Anguelov, C. Dulong, D. Filip, C. Frueh, S. Lafon, R. Lyon, A. Ogale, L. Vincent, and J. Weaver. Google Street View: Capturing the World at Street Level. *Computer*, 43(6):32–38, 2010. 87

[7] C. Appert and M. Beaudouin-Lafon. SwingStates: Adding State Machines to Java and the Swing Toolkit. *Software: Practice and Experience*, 38(11):1149–1182, 2008. 37, 38

[8] M. Araki and K. Tachibana. Multimodal Dialog Description Language for Rapid System Development. In *Proceedings of the 7th SIGdial Workshop on Discourse and Dialogue*, pages 109–116, 2009. 36

[9] M. Arikawa, S. Konomi, and K. Ohnishi. Navitime: Supporting Pedestrian Navigation in the Real World. *IEEE Pervasive Computing*, 6(3):21–29, 2007. 35

[10] N. Armstrong, C. Nugent, G. Moore, and D. Finlay. Developing Smartphone Applications for People with Alzheimer's Disease. In *10th IEEE International Conference on Information Technology and Applications in Biomedicine (ITAB)*, pages 1–5. IEEE, 2010. 24

[11] M. T. I. Aumi, S. Gupta, M. Goel, E. Larson, and S. Patel. DopLink: Using the Doppler Effect for Multi-Device Interaction. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, pages 583–586. ACM, 2013. 32

[12] R. Azuma. A Survey of Augmented Reality. *Presence-Teleoperators and Virtual Environments*, 6(4):355–385, 1997. 35

[13] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre. Recent Advances in Augmented Reality. *IEEE Computer Graphics and Applications*, 21(6):34–47, 2001. 35

[14] M. Bähr, S. Klein, S. Diewald, C. Haag, G. Hofstetter, M. Khoury, D. Kurz, A. Winkler, A. König, N. Holzer, M. Siegrist, A. Pressler, L. Roalter, T. Linner, K. Wessig, M. Heuberger, V. Warmuth, M. Kranz, and T. Bock. PASSAge – Personalized Mobility, Assistance and Service Systems in an Ageing Society. In *6. Deutscher AAL-Kongress*, pages 260–269. VDE Verlag, 2013. 68

[15] A. Bangor, P. Kortum, and J. Miller. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of Usability Studies*, 4(3):114–123, 2009. 56, 145, 150

[16] L. Bao and S. Intille. Activity Recognition from User-Annotated Acceleration Data. In A. Ferscha and F. Mattern, editors, *Pervasive Computing*, volume 3001 of *Lecture Notes in Computer Science*, pages 1–17. Springer Berlin/Heidelberg, 2004. 25, 70

[17] J. E. Bardram, M. Frost, K. Szántó, and G. Marcu. The MONARCA Self-Assessment System: A Persuasive Personal Monitoring System for Bipolar Patients. In *Proceedings of the 2nd ACM SIGHIT International Health Informatics Symposium*, pages 21–30. ACM, 2012. 24

[18] L. Barkhuus and A. K. Dey. Location-Based Services for Mobile Telephony: A Study of Users' Privacy Concerns. In *Proceedings of the International Conference on Human-Computer Interaction (INTERACT)*, pages 702–712, 2003. 40

[19] L. Barkhuus and V. E. Polichar. Empowerment through Seamfulness: Smart Phones in Everyday Life. *Personal and Ubiquitous Computing*, 15(6):629–639, 2010. 165

[20] J. Baus, K. Cheverst, and C. Kray. A Survey of Map-Based Mobile Guides. In *Map-based Mobile Services*, pages 193–209. Springer, 2005. 34

[21] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded Up Robust Features. In *Proceedings of the 9th European Conference on Computer Vision (ECCV)*, pages 404–417. Springer, 2006. 32

[22] A. K. Beeharee and A. Steed. A Natural Wayfinding Exploiting Photos in Pedestrian Navigation Systems. In *Proceedings of the 8th Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI)*, pages 81–88. ACM, 2006. 35

[23] J. Berg, L. Berquam, and K. Christoph. Social Networking Technologies: A "Poke" for Campus Services. *Educause Review*, 42(2), 2007. 30

[24] N. O. Bernsen. Multimodality in Language and Speech Systems – From Theory to Design Support Tool. In B. Granström, D. House, and I. Karlsson, editors, *Multimodality in Language and Speech Systems*, volume 19 of *Text, Speech and Language Technology*, pages 93–148. Springer Netherlands, 2002. 18, 19, 20

[25] M. Böhmer, B. Hecht, J. Schöning, A. Krüger, and G. Bauer. Falling Asleep with Angry Birds, Facebook and Kindle: A Large Scale Study on Mobile Application Usage. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI)*, pages 47–56. ACM, 2011. 41, 165, 175, 181

[26] M. Böhmer and A. Krüger. A Case Study of Research through the App Store: Leveraging the System UI as a Playing Field for Improving the Design of Smartphone Launchers. *International Journal of Human-Computer Interaction*, 2014. 175

[27] N. Bolger, A. Davis, and E. Rafaeli. Diary Methods: Capturing Life as it is Lived. *Annual Review of Psychology*, 54:579–616, 2003. 42

[28] S. Boll, N. Henze, M. Pielot, B. Poppinga, and T. Schinke. My App is an Experiment: Experience from User Studies in Mobile App Stores. *International Journal of Mobile Human-Computer Interaction (IJMHCI)*, 3(4):71–91, 2011. 42, 174, 176

[29] R. A. Bolt. "Put-that-there": Voice and Gesture at the Graphics Interface. In *Proceedings of the 7th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 262–270. ACM, 1980. 18

[30] S. Boring, D. Baur, A. Butz, S. Gustafson, and P. Baudisch. Touch Projector: Mobile Interaction Through Video. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 2287–2296. ACM, 2010. 18, 39

[31] S. Bosman, B. Groenendaal, J. Findlater, T. Visser, M. de Graaf, and P. Markopoulos. GentleGuide: An Exploration of Haptic Output for Indoors Pedestrian Guidance. In L. Chittaro, editor, *Human-Computer Interaction with Mobile Devices and Services*, volume 2795 of *Lecture Notes in Computer Science*, pages 358–362. Springer Berlin Heidelberg, 2003. 35

[32] M.-L. Bourguet. A Toolkit for Creating and Testing Multimodal Interface Designs. In *Proceedings of the ACM User Interface Software and Technology Symposium (UIST)*, pages 29–30. ACM, 2002. 37

[33] M.-L. Bourguet. Designing and Prototyping Multimodal Commands. In *Proceedings of the International Conference on Human-Computer Interaction (INTERACT)*, pages 717–720, 2003. 9

[34] J. Brooke. SUS – A Quick and Dirty Usability Scale. *Usability Evaluation in Industry*, pages 189–194, 1996. 55, 57, 144, 150

[35] K. A. Bruffee. *Collaborative Learning: Higher Education, Interdependence, and the Authority of Knowledge*. The Johns Hopkins University Press, 2nd edition, 1998. 78

[36] A. J. B. Brush. Ubiquitous Computing Field Studies. In J. Krumm, editor, *Ubiquitous Computing Fundamentals*. Chapman & Hall / CRC, 2010. 39, 40, 41, 42

[37] M. Buhrmester, T. Kwang, and S. D. Gosling. Amazon's Mechanical Turk: A New Source of Inexpensive, Yet High-Quality, Data? *Perspectives on Psychological Science*, 6(1):3–5, 2011. 174, 175

[38] H. Bunt. Issues in Multimodal Human-Computer Communication. In H. Bunt, R.-J. Beun, and T. Borghuis, editors, *Multimodal Human-Computer Communication*, volume

1374 of *Lecture Notes in Computer Science*, pages 1–12. Springer Berlin Heidelberg, 1998. 21

[39] S. Burbeck. Applications Programming in Smalltalk-80(TM): How to use Model-View-Controller (MVC). Technical report, Department of Computer Science, University of Illinois, USA, 1992. 100

[40] A. Butz, J. Baus, A. Krüger, and M. Lohse. A Hybrid Indoor Navigation System. In *Proceedings of the 6th International Conference on Intelligent User Interfaces (IUI)*, pages 25–32. ACM, 2001. 31, 34

[41] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary Robust Independent Elementary Features. In *Proceedings of the 13th European Conference on Computer Vision (ECCV)*, pages 778–792. Springer, 2010. 32

[42] A. Campbell, T. Choudhury, S. Hu, H. Lu, M. K. Mukerjee, M. Rabbi, and R. D. Raizada. NeuroPhone: Brain-mobile Phone Interface Using a Wireless EEG Headset. In *Proceedings of the 2nd ACM SIGCOMM Workshop on Networking, Systems, and Applications on Mobile Handhelds (MobiHeld)*, pages 3–8. ACM, 2010. 17

[43] J. Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (6):679–698, 1986. 99

[44] S. Carter and J. Mankoff. When Participants Do the Capturing: The Role of Media in Diary Studies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 899–908. ACM, 2005. 41

[45] S. Carter, J. Mankoff, and J. Heer. Momento: Support for Situated Ubicomp Experimentation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 125–134. ACM, 2007. 42

[46] D. Chang, L. Dooley, and J. E. Tuovinen. Gestalt Theory in Visual Screen Design: A New Look at an Old Subject. In *Proceedings of the 7th World Conference on Computers in Education: Australian Topics*, volume 8, pages 5–12. Australian Computer Society, 2002. 19

[47] K.-H. Chang, M. Y. Chen, and J. Canny. Tracking Free-Weight Exercises. In *Proceedings of the 9th International Conference on Ubiquitous Computing (UbiComp)*, pages 19–37. Springer, 2007. 25

[48] M. A. Chaqfeh and N. Mohamed. Challenges in Middleware Solutions for the Internet of Things. In *International Conference on Collaboration Technologies and Systems (CTS)*, pages 21–26. IEEE, 2012. 136

[49] K. Cheverst, N. Davies, A. Friday, and C. Efstratiou. Developing a Context-Aware Electronic Tourist Guide: Some Issues and Experiences. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 17–24. ACM, 2000. 30, 31

[50] L. Chittaro. Distinctive Aspects of Mobile Interaction and their Implications for the Design of Multimodal Interfaces. *Journal on Multimodal User Interfaces*, 3(3):157–165, 2010. 22, 23

[51] T. C. Christensen, L. F. Barrett, E. Bliss-Moreau, K. Lebo, and C. Kaschub. A Practical Guide to Experience-Sampling Procedures. *Journal of Happiness Studies*, 4(1):53–78, 2003. 41

[52] W. Cochran and G. Cox. *Experimental Designs*. John Wiley & Sons, 1957. 54

[53] T. Cochrane, V. Narayan, and J. Oldfield. iPadagogy: Appropriating the iPad Within Pedagogical Contexts. *International Journal of Mobile Learning and Organisation (IJMLO)*, 7(1):48–65, 2013. 28

[54] P. R. Cohen, M. Johnston, D. McGee, S. L. Oviatt, J. Clow, and I. A. Smith. The Efficiency of Multimodal Interaction: A Case Study. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, 1998. 40

[55] M. Colbert. A Diary Study of Rendezvousing: Implications for Position-Aware Computing and Communications for the General Public. In *Proceedings of the International ACM SIGGROUP Conference on Supporting Group Work*, pages 15–23. ACM, 2001. 42

[56] S. Consolvo, B. Harrison, I. Smith, M. Chen, K. Everitt, J. Froehlich, and J. A. Landay. Conducting In Situ Evaluations for and with Ubiquitous Computing Technologies. *International Journal of Human-Computer Interaction*, 22(1–2):103–118, 2007. 41

[57] S. Consolvo, D. W. McDonald, T. Toscos, M. Y. Chen, J. Froehlich, B. Harrison, P. Klasnja, A. LaMarca, L. LeGrand, R. Libby, I. Smith, and J. A. Landay. Activity Sensing in the Wild: a Field Trial of Ubifit Garden. In *Proceeding of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 1797–1806. ACM, 2008. 26, 27

[58] S. Consolvo, P. Roessler, B. E. Shelton, A. LaMarca, B. Schilit, and S. Bly. Technology for Care Networks of Elders. *IEEE Pervasive Computing*, 3(2):22–29, 2004. 24

[59] N. R. Cook and J. H. Ware. Design and Analysis Methods for Longitudinal Research. *Annual Review of Public Health*, 4:1–23, 1983. 40

[60] A. Cooper. *The Inmates are Running the Asylum: Why High-Tech Products Drive Us Crazy and How to Restore the Sanity*. Sams Indianapolis, 1999. 82

[61] J. F. Coppola and B. A. Thomas. Beyond "Chalk and Talk": A Model for E-Classroom Design. *THE Journal*, 27(6):30–32, 2000. 76

[62] J. Coutaz. Multimedia and Multimodal User Interfaces: A Taxonomy for Software Engineering Research Issues. In *St. Petersburg HCI Workshop*, 1992. 10

[63] J. Coutaz, L. Nigay, D. Salber, A. Blandford, J. May, and R. M. Young. Four Easy Pieces for Assessing the Usability of Multimodal Interaction: The Care Properties. In *Proceedings of the International Conference on Human-Computer Interaction (INTERACT)*, volume 95, pages 115–120, 1995. 36

[64] H. Cramer, M. Rost, N. Belloni, F. Bentley, and D. Chincholle. Research in the Large. Using App Stores, Markets, and Other Wide Distribution Channels in Ubicomp Research. In *Adjunct Proceedings of the 12th ACM International Conference on Ubiquitous Computing (UbiComp)*, pages 511–514. ACM, 2010. 42, 174

[65] H. Cramer, M. Rost, and F. Bentley. An Introduction to Research in the Large, Guest

Editorial Preface. *Special Issue of International Journal of Mobile Human-Computer Interaction (IJMHCI)*, 2011. 174

[66] F. Cutugno, V. A. Leano, R. Rinaldi, and G. Mignini. Multimodal Framework for Mobile Interaction. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, pages 197–203. ACM, 2012. 37, 38

[67] M. Czerwinski, E. Horvitz, and S. Wilhite. A Diary Study of Task Switching and Interruptions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 175–182. ACM, 2004. 42

[68] S. da Costa Ribeiro, M. Kleinsteuber, A. Möller, and M. Kranz. Data Acquisition for Motion Recognition on Mobile Platforms via Compressive Sensing. In *Proceedings of the 13th International Conference on Computer Aided Systems Theory (EUROCAST)*, pages 388–389, 2011. 25

[69] S. da Costa Ribeiro, M. Kleinsteuber, A. Möller, and M. Kranz. A Compressive Sensing Scheme of Frequency Sparse Signals for Mobile and Wearable Platforms. In R. Moreno-Díaz, F. Pichler, and A. Quesada-Arencibia, editors, *Computer Aided Systems Theory – EUROCAST 2011*, volume 6928 of *Lecture Notes in Computer Science*, pages 510–518. Springer Berlin Heidelberg, 2012. 25

[70] S. Daukantas, V. Marozas, and A. Lukosevicius. Inertial Sensor for Objective Evaluation of Swimmer Performance. In *11th International Biennial Baltic Electronics Conference (BEC)*, pages 321–324. IEEE, 2008. 26

[71] L. Dayer, S. Heldenbrand, P. Anderson, P. O. Gubbins, and B. C. Martin. Smartphone Medication Adherence Apps: Potential Benefits to Patients and Providers. *Journal of the American Pharmacists Association (JAPhA)*, 53(2):172–181, 2013. 24

[72] F. P. Deane, J. Podd, and R. D. Henderson. Relationship between Self-Report and Log Data Estimates of Information System Usage. *Computers in Human Behavior*, 14(4):621–636, 1998. 168

[73] G. Dedes and A. G. Dempster. Indoor GPS Positioning. In *Proceedings of the IEEE Semiannual Vehicular Technology Conference*. IEEE, 2005. 31

[74] R. Demumieux and P. Losquin. Gather Customer's Real Usage on Mobile Phones. In *Proceedings of the 7th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI)*, pages 267–270. ACM, 2005. 165

[75] T. Denning, A. Andrew, R. Chaudhri, C. Hartung, J. Lester, G. Borriello, and G. Duncan. BALANCE: Towards a Usable Pervasive Wellness Application with Accurate Activity Inference. In *Proceedings of the 10th Workshop on Mobile Computing Systems and Applications (HotMobile)*, pages 1–5. ACM, 2009. 26

[76] A. K. Dey, G. D. Abowd, and D. Salber. A Conceptual Framework and a Toolkit for Supporting the Rapid Prototyping of Context-Aware Applications. *Human-Computer Interaction*, 16(2):97–166, 2001. 36

[77] S. Diewald, A. Möller, L. Roalter, and M. Kranz. Mobile Device Integration and Interaction in the Automotive Domain. In *Proceedings of AutoNUI: Automotive Natural User*

Interfaces Workshop at the 3rd International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI), 2011. 22

[78] S. Diewald, A. Möller, L. Roalter, and M. Kranz. DriveAssist – A V2X-Based Driver Assistance System for Android. In *Mensch & Computer Workshopband*, pages 373–380, 2012. 22

[79] S. Diewald, A. Möller, L. Roalter, and M. Kranz. Gamification-Supported Exploration of Natural User Interfaces. In *Adjunct Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI)*, pages 47–48, 2012. 27, 46

[80] S. Diewald, A. Möller, L. Roalter, and M. Kranz. MobiliNet: A Social Network for Optimized Mobility. In *Adjunct Proceedings of the 4th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (AutomotiveUI)*, pages 145–150, 2012. 119

[81] S. Diewald, A. Möller, L. Roalter, and M. Kranz. Simulation and Digital Prototyping of Tangible User Interfaces. In *Proceedings of Mensch & Computer 2014: Interaktiv unterwegs – Freiräume gestalten*, pages 371–374. Oldenbourg Verlag, 2014. 134

[82] S. Diewald, A. Möller, L. Roalter, T. Stockinger, and M. Kranz. Gameful Design in the Automotive Domain: Review, Outlook and Challenges. In *Proceedings of the 5th International Conference on Automotive User Interfaces and Interactive Vehicular Applications (Automotive UI)*, pages 262–265. ACM, 2013. 27

[83] S. Diewald, A. Möller, T. Stockinger, L. Roalter, M. Koelle, P. Lindemann, and M. Kranz. Gamification-Supported Exploration and Practicing for Automotive User Interfaces and Vehicle Functions. In T. Reiners and L. C. Wood, editors, *Gamification in Education and Business*. Springer, 2014. 27

[84] S. Diewald, L. Roalter, A. Möller, and M. Kranz. Towards a Holistic Approach for Mobile Application Development in Intelligent Environments. In *Proceedings of the 10th International Conference on Mobile and Ubiquitous Multimedia (MUM)*, pages 73–80. ACM, 2011. 24, 71

[85] S. Diewald, L. Roalter, A. Möller, and M. Kranz. Simulation of Tangible User Interfaces with the ROS Middleware. In *Adjunct Proceedings of the 8th International Conference on Tangible, Embedded and Embodied Interaction*, 2014. 134, 136

[86] G. Dragana, R. Dragica, K. Dijana, and I. Dragica. Pedagogical and Didactic-Methodical Aspects of E-Learning. In *Proceedings of the 6th WSEAS International Conference on E-Activities*, pages 65–73, 2007. 29

[87] Y. Du, Y. Chen, D. Wang, J. Liu, and Y. Lu. An Android-Based Emergency Alarm and Healthcare Management System. In *International Symposium on IT in Medicine and Education (ITME)*, pages 375–379, 2011. 24

[88] B. Dumas, D. Lalanne, and R. Ingold. Description Languages for Multimodal Interaction: A Set of Guidelines and its Illustration with SMUIML. *Journal on Multimodal User Interfaces*, 3(3):237–247, 2010. 36

[89] B. Dumas, D. Lalanne, and S. Oviatt. Multimodal Interfaces: A Survey of Principles, Models and Frameworks. In D. Lalanne and J. Kohlas, editors, *Human Machine Interaction*, volume 5440 of *Lecture Notes in Computer Science*, pages 3–26. Springer Berlin Heidelberg, 2009. 23, 37

[90] O. J. Dunn. Multiple Comparisons among Means. *Journal of the American Statistical Association*, 56(293):52–64, 1961. 93

[91] N. Eagle, A. S. Pentland, and D. Lazer. Inferring Friendship Network Structure by Using Mobile Phone Data. *Proceedings of the National Academy of Sciences*, 106(36):15274–15278, 2009. 41

[92] N. Elouali, X. Le Pallec, J. Rouillard, and J.-C. Tarby. MIMIC: Leveraging Sensor-Based Interactions in Multimodal Mobile Applications. In *Extended Abstracts on Human Factors in Computing Systems (CHI)*, pages 2323–2328. ACM, 2014. 4, 36, 38

[93] S. Erdt, T. Linner, L. Herdener, J. Riess, M. Kreitmaier, L. Roalter, T. Schulz, M. Struck, W. Setz, T. Bock, M. Kranz, V. Velioglu, and E. Moritz. Systematische Entwicklung eines komplexen multidimensionalen Assistenzsystems am Beispiel des GEWOS-Bewegungssessels. In *Tagungsband zum 5. Deutschen AAL Kongress 2012*. VDE-Verlag, 2012. 70

[94] M. Ermes, J. Parkka, J. Mantyjarvi, and I. Korhonen. Detection of Daily Activities and Sports with Wearable Sensors in Controlled and Uncontrolled Conditions. *IEEE Transactions on Information Technology in Biomedicine*, 12(1):20–26, 2008. 25, 70

[95] H. Falaki, R. Mahajan, S. Kandula, D. Lymberopoulos, R. Govindan, and D. Estrin. Diversity in Smartphone Usage. In *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 179–194. ACM, 2010. 41, 165

[96] N. Fallah, I. Apostolopoulos, K. Bekris, and E. Folmer. Indoor Human Navigation Systems: A Survey. *Interacting with Computers*, 25(1):21–33, 2013. 31

[97] M. Fayad and D. C. Schmidt. Object-Oriented Application Frameworks. *Communications of the ACM*, 40(10):32–38, 1997. 36

[98] D. Ferreira. *AWARE: A Mobile Context Instrumentation Middleware to Collaboratively Understand Human Behavior*. PhD thesis, University of Oulu, Finland, 2013. 42

[99] D. Ferreira, A. K. Dey, and V. Kostakos. Understanding Human-Smartphone Concerns: A Study of Battery Life. In *Proceedings of the 9th International Conference on Pervasive Computing (Pervasive)*, pages 19–33. Springer, 2011. 175, 176

[100] L. D. Fink. *Creating Significant Learning Experiences: An Integrated Approach to Designing College Courses*. John Wiley & Sons, 2013. 75

[101] F. Flippo, A. Krebs, and I. Marsic. A Framework for Rapid Development of Multimodal Interfaces. In *Proceedings of the 5th International Conference on Multimodal Interfaces (ICMI)*, pages 109–116. ACM, 2003. 37

[102] J. Fontecha, F. J. Navarro, R. Hervás, and J. Bravo. Elderly Frailty Detection by Using

Accelerometer-Enabled Smartphones and Clinical Information Records. *Personal and Ubiquitous Computing*, 17(6):1073–1083, 2013. 24

[103] S. Frederick, G. Loewenstein, and T. O'Donoghue. Time Discounting and Time Preference: A Critical Review. *Journal of Economic Literature*, 40(2):351–401, 2002. 45

[104] M. Friedman. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *Journal of the American Statistical Association*, 32(200):675–701, 1937. 93

[105] G. Fritz, C. Seifert, and L. Paletta. A Mobile Vision System for Urban Detection with Informative Local Descriptors. In *IEEE International Conference on Computer Vision Systems (ICVS)*, pages 30–38. IEEE, 2006. 31

[106] 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 (MobiSys)*, pages 57–70. ACM, 2007. 41, 42

[107] M. Fukumoto. A Finger-Ring Shaped Wearable Handset Based on Bone-Conduction. In *Proceedings of the 9th IEEE International Symposium on Wearable Computers*, pages 10–13. IEEE, 2005. 17

[108] R. K. Ganti, F. Ye, and H. Lei. Mobile Crowdsensing: Current State and Future Challenges. *IEEE Communications Magazine,*, 49(11):32–39, 2011. 174

[109] J. Gao and A. Koronios. Mobile Application Development for Senior Citizens. In *Proceedings of the Pacific Asia Conference on Information Systems (PACIS)*, 2010. Paper 65. 24

[110] R. Garrison. Theoretical Challenges for Distance Education in the 21^{st} Century: A Shift from Structural to Transactional Issues. *The International Review of Research in Open and Distance Learning*, 1(1), 2000. 28

[111] B. Geilhof, J. Güttler, M. Heuberger, S. Diewald, and D. Kurz. Weiterentwicklung existierender Assistenz- und Mobilitätshilfen für Senioren – Nutzen, Akzeptanz und Potenziale. In G. Kempter and W. Ritter, editors, *uDay XII – Assistenztechnik für betreutes Wohnen*. Pabst Science Publisher, 2014. 25

[112] T. Georgiev, Z. Yu, A. Lumsdaine, and S. Goma. Lytro Camera Technology: Theory, Algorithms, Performance Analysis. In *Proceedings of the SPIE*, volume 8867, pages 1–10. International Society for Optics and Photonics, 2013. 13

[113] A. Hang, G. Broll, and A. Wiethoff. Visual Design of Physical User Interfaces for NFC-Based Mobile Interaction. In *Proceedings of the 8th ACM Conference on Designing Interactive Systems (DIS)*, pages 292–301. ACM, 2010. 39

[114] A. Hang, E. Rukzio, and A. Greaves. Projector Phone: A Study of Using Mobile Phones with Integrated Projector for Interaction with Maps. In *Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI)*, pages 207–216. ACM, 2008. 16

[115] J. Harding, J. Small, and D. James. Feature Extraction of Performance Variables in Elite Half-Pipe Snowboarding Using Body Mounted Inertial Sensors. In *Proceedings of SPIE, BioMEMS and Nanotechnology III*, volume 6799, 2007. 26

[116] R. Hardy and E. Rukzio. Touch & Interact: Touch-Based Interaction of Mobile Phones with Displays. In *Procedings of the 10th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI)*, pages 245–254. ACM, 2008. 39

[117] C. Harrison and S. E. Hudson. Scratch Input: Creating Large, Inexpensive, Unpowered and Mobile Finger Input Surfaces. In *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology (UIST)*, pages 205–208. ACM, 2008. 18

[118] C. Harrison, D. Tan, and D. Morris. Skinput: Appropriating the Body as an Input Surface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 453–462. ACM, 2010. 17

[119] C. Hartley, M. Brecht, P. Pagerey, G. Weeks, A. Chapanis, and D. Hoecker. Subjective Time Estimates of Work Tasks by Office Workers. *Journal of Occupational Psychology*, 50(1):23–36, 2011. 168

[120] T. L. Hayes, K. Cobbinah, T. Dishongh, J. A. Kaye, J. Kimel, M. Labhard, T. Leen, J. Lundell, U. Ozertem, and M. Pavel. A Study of Medication-Taking and Unobtrusive, Intelligent Reminding. *Telemedicine and e-Health*, 15(8):770–776, 2009. 47

[121] S. Heim. *The Resonant Interface: HCI Foundations for Interaction Design*. Addison-Wesley Longman Publishing Co., Inc., 2007. 10

[122] M. G. Helander, T. K. Landauer, and P. V. Prabhu. *Handbook of Human-Computer Interaction*. Elsevier, 1997. 118

[123] N. Henze, E. Rukzio, and S. Boll. 100,000,000 Taps: Analysis and Improvement of Touch Performance in the Large. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI)*, pages 133–142. ACM, 2011. 42, 175, 176

[124] N. Henze, E. Rukzio, and S. Boll. Observational and Experimental Investigation of Typing Behaviour Using Virtual Keyboards for Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 2659–2668. ACM, 2012. 42, 175, 176

[125] H. Hile and G. Borriello. Positioning and Orientation in Indoor Environments Using Camera Phones. *IEEE Computer Graphics and Applications*, 28(4):32–39, 2008. 33

[126] H. Hile, R. Vedantham, G. Cuellar, A. Liu, N. Gelfand, R. Grzeszczuk, and G. Borriello. Landmark-Based Pedestrian Navigation from Collections of Geotagged Photos. In *Proceedings of the 7th International Conference on Mobile and Ubiquitous Multimedia (MUM)*, pages 145–152. ACM, 2008. 35

[127] S. Hilsenbeck, D. Bobkov, G. Schroth, R. Huitl, and E. Steinbach. Graph-based Data Fusion of Pedometer and WiFi Measurements for Mobile Indoor Positioning. In *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp)*, 2014. 31

[128] S. Hilsenbeck, A. Möller, R. Huitl, G. Schroth, M. Kranz, and E. Steinbach. Scale-Preserving Long-Term Visual Odometry for Indoor Navigation. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–10. IEEE, 2012. 31, 107

[129] K. Hinckley and H. Song. Sensor Synaesthesia: Touch in Motion, and Motion in Touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 801–810. ACM, 2011. 18

[130] S. Hodges, E. Berry, and K. Wood. SenseCam: A Wearable Camera that Stimulates and Rehabilitates Autobiographical Memory. *Memory*, 19(7):685–696, 2011. 24

[131] S. Hoeffler. Measuring Preferences for Really New Products. *Journal of Marketing Research*, 40(4):406–420, 2003. 20

[132] P. Holleis, M. Kranz, and A. Schmidt. Displayed Connectivity. In *Adjunct Proceedings of the 7th International Conference on Ubiquitous Computing (UbiComp)*, 2005. 30

[133] P. Holleis, M. Kranz, A. Winter, and A. Schmidt. Playing with the Real World. *Journal of Virtual Reality and Broadcasting*, 3(1):1–12, 2006. 29, 71

[134] A. Holzinger, A. Nischelwitzer, and M. Kickmeier-Rust. Pervasive E-Education Supports Life Long Learning: Some Examples of X-Media Learning Objects. In *Proceedings of the 10th IACEE World Conference on Continuing Engineering Education (WCCEE*, 2008. 28, 29

[135] A. Holzinger, A. Nischelwitzer, and M. Meisenberger. Mobile Phones as a Challenge for m-Learning: Examples for Mobile Interactive Learning Objects (MILOs). In *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW)*, pages 307–311. IEEE, 2005. 28

[136] X. Hou and L. Zhang. Saliency Detection: A Spectral Residual Approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE, 2007. 19

[137] M. R. Hufford and A. L. Shields. Electronic Diaries: Applications and What Works in the Field. *Applied Clinical Trials*, pages 46–59, 2002. 42

[138] R. Huitl, G. Schroth, S. Hilsenbeck, F. Schweiger, and E. Steinbach. TUMindoor: An Extensive Image and Point Cloud Dataset for Visual Indoor Localization and Mapping. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*. IEEE, 2012. 33

[139] S. Hulkko, T. Mattelmäki, K. Virtanen, and T. Keinonen. Mobile Probes. In *Proceedings of the 3rd Nordic Conference on Human-Computer Interaction (NordiCHI)*, pages 43–51. ACM, 2004. 41, 42

[140] T. Ishigaki, T. Higuchi, and K. Watanabe. Spectrum Classification for Early Fault Diagnosis of the LP Gas Pressure Regulator Based on the Kullback-Leibler Kernel. In *Proceedings of the 16th IEEE Signal Processing Society Workshop*, pages 453–458, 2006. 64

[141] I. Jacobson, G. Booch, and J. Rumbaugh. *The Unified Software Development Process*, volume 1. Addison-Wesley Reading, 1999. 6

[142] I. Jahnke, P. Bergström, K. Lindwall, E. Mårell-Olsson, A. Olsson, F. Paulsson, and P. Vinnervik. Understanding, Reflecting and Designing Learning Spaces of Tomorrow. In *Proceedings of The IADIS International Conference "Mobile Learning 2012"*, pages 147–156, 2012. 28

[143] A. Jaimes and N. Sebe. Multimodal Human-Computer Interaction: A Survey. *Computer Vision and Image Understanding*, 108(1):116–134, 2007. 9

[144] B. J. Jansen, I. Taksa, and A. Spink. *Handbook of Research on Web Log Analysis*. IGI Global, 2009. 41

[145] M. H. Jeon, D. Y. Na, J. H. Ahn, and J. Y. Hong. User Segmentation & UI Optimization Through Mobile Phone Log Analysis. In *Proceedings of the 10th International Conference on Human Computer Interaction with Mobile Devices and Services (MobileHCI)*, pages 495–496. ACM, 2008. 165

[146] K. Jokinen and T. Hurtig. User Expectations and Real Experience on a Multimodal Interactive System. In *Proceedings of the 9th International Conference on Spoken Language Processing (INTERSPEECH)*, 2006. 21

[147] T. Kallio and A. Kaikkonen. Usability Testing of Mobile Applications: A Comparison between Laboratory and Field Testing. *Journal of Usability Studies*, 1(4-16):23–28, 2005. 160

[148] S. Kang, J. Lee, H. Jang, H. Lee, Y. Lee, S. Park, T. Park, and J. Song. SeeMon: Scalable and Energy-Efficient Context Monitoring Framework for Sensor-Rich Mobile Environments. In *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 267–280. ACM, 2008. 36, 37

[149] T. Kärkkäinen, T. Vaittinen, and K. Väänänen-Vainio-Mattila. I Don't Mind Being Logged, But Want to Remain in Control: A Field Study of Mobile Activity and Context Logging. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 163–172. ACM, 2010. 41

[150] K. Katsurada, Y. Nakamura, H. Yamada, and T. Nitta. XISL: A Language for Describing Multimodal Interaction Scenarios. In *Proceedings of the 5th International Conference on Multimodal Interfaces (ICMI)*, pages 281–284. ACM, 2003. 36

[151] N. Keijsers, M. Horstink, and S. Gielen. Ambulatory Motor Assessment in Parkinson's Disease. *Movement Disorders*, 21(1):34–44, 2006. 25

[152] J. F. Kelley. An Empirical Methodology For Writing User-Friendly Natural Language Computer Applications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 193–196. ACM, 1983. xvi, 39, 91, 100, 157

[153] C. Kiefer, N. Collins, and G. Fitzpatrick. HCI Methodology for Evaluating Musical Controllers: A Case Study. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, 2008. 40

[154] E. Kiel. Strukturierung. In E. Kiel, editor, *Unterricht sehen, analysieren, gestalten*, pages 21–36. Klinkhardt, 2nd edition, 2008. 75

[155] S. H. Kim, C. Mims, and K. P. Holmes. An Introduction to Current Trends and Benefits of Mobile Wireless Technology Use in Higher Education. *AACE Journal*, 14(1):77–100, 2006. 28

[156] A. Kittur, E. H. Chi, and B. Suh. Crowdsourcing User Studies with Mechanical Turk. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 453–456. ACM, 2008. 174, 175

[157] P. Klasnja and W. Pratt. Managing Health with Mobile Technology. *ACM interactions*, 21(1):66–69, 2014. 45, 46

[158] G. Klein and D. Murray. Parallel Tracking and Mapping on a Camera Phone. In *Proceedings of the 8th IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, 2009. 31, 90

[159] A. Kolmogorov. Sulla Determinazione Empirica di una Legge di Distribuzione. *G. Ist. Ital. Attuari*, pages 83–91, 1933. 112

[160] P. Korpipää, J. Häkkilä, J. Kela, S. Ronkainen, and I. Känsälä. Utilising Context Ontology in Mobile Device Application Personalisation. In *Proceedings of the 3rd International Conference on Mobile and Ubiquitous Multimedia (MUM)*, pages 133–140. ACM, 2004. 36

[161] S. B. Kotsiantis. Supervised Machine Learning: A Review of Classification Techniques. In *Proceedings of the Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, pages 3–24. IOS Press, 2007. 141

[162] N. Krahnstoever, S. Kettebekov, M. Yeasin, and R. Sharma. A Real-Time Framework for Natural Multimodal Interaction with Large Screen Displays. In *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces (ICMI)*, page 349. IEEE, 2002. 37

[163] M. Kranz. *Engineering Perceptive User Interfaces*. PhD thesis, Ludwig-Maximilians-Universität München, Germany, 2008. 6

[164] M. Kranz, C. Fischer, and A. Schmidt. A Comparative Study of DECT and WLAN Signals for Indoor Localization. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 235–243. IEEE, 2010. 15, 31, 78

[165] M. Kranz, P. Holleis, and A. Schmidt. DistScroll – A New One-Handed Interaction Device. In *Proceedings of the 25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW)*, pages 499–505. IEEE, 2005. 17

[166] M. Kranz, P. Holleis, and A. Schmidt. Ubiquitous Presence Systems. In *Proceedings of the ACM Symposium on Applied Computing (SAC)*, pages 1902–1909. ACM, 2006. 82

[167] M. Kranz, P. Holleis, W. Spiessl, and A. Schmidt. The Therapy Top Measurement and Visualization System – An Example for the Advancements in Existing Sports Equipments. *International Journal of Computer Science in Sport*, 5(2):76–80, 2006. 25, 60

[168] M. Kranz, A. Möller, S. Diewald, L. Roalter, B. Beege, B. Meyer, and A. Hendrich. Mobile and Contextual Learning: A Case Study on Mobile Didactics in Teaching and Education. *International Journal of Mobile Learning and Organisation*, 7(2):113–139, Aug. 2013. 72, 73, 80, 81, 160

[169] M. Kranz, A. Möller, N. Hammerla, S. Diewald, T. Plötz, P. Olivier, and L. Roalter. The Mobile Fitness Coach: Towards Individualized Skill Assessment Using Personalized Mobile Devices. *Pervasive and Mobile Computing*, 9(2):203–215, 2013. 26, 39, 46, 62, 66

[170] M. Kranz, A. Möller, and L. Roalter. Robots, Objects, Humans: Towards Seamless Interaction in Intelligent Environments. In *1st International Conference on Pervasive and Embedded Computing and Communication Systems (PECCS)*, pages 163–172. SciTePress, 2011. 46

[171] M. Kranz, L. Murmann, and F. Michahelles. Research in the Large: Challenges for Large-Scale Mobile Application Research – A Case Study about NFC Adoption using Gamification via an App Store. *International Journal of Mobile Human-Computer Interaction (IJMHCI)*, 5(1):45–61, 2013. 42, 46, 176

[172] M. Kranz, A. Schmidt, R. B. Rusu, A. Maldonado, M. Beetz, B. Hörnler, and G. Rigoll. Sensing Technologies and the Player-Middleware for Context-Awareness in Kitchen Environments. In *Proceedings of the 4th International Conference on Networked Sensing Systems (INSS)*, pages 179–186, 2007. 24, 29

[173] C. Kray, C. Elting, K. Laakso, and V. Coors. Presenting Route Instructions on Mobile Devices. In *Proceedings of the 8th International Conference on Intelligent User Interfaces (IUI)*, pages 117–124. ACM, 2003. 33, 34

[174] C. Kray and G. Kortuem. Interactive Positioning Based on Object Visibility. In S. Brewster and M. Dunlop, editors, *Proceedings of the 6th Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI)*, volume 3160 of *Lecture Notes in Computer Science*, pages 276–287. Springer Berlin Heidelberg, 2004. 89

[175] N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell. A Survey of Mobile Phone Sensing. *IEEE Communications Magazine*, 48(9):140–150, 2010. 70

[176] O. D. Lara and M. A. Labrador. A Survey on Human Activity Recognition Using Wearable Sensors. *IEEE Communications Surveys & Tutorials*, 15(3):1192–1209, 2013. 11, 12

[177] J. Lazar, J. H. Feng, and H. Hochheiser. *Research Methods in Human-Computer Interaction*. John Wiley & Sons Ltd, Chichester, 2010. 39, 40, 41

[178] D. A. Lazewatsky and W. D. Smart. An Inexpensive Robot Platform for Teleoperation and Experimentation. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1211–1216. IEEE, 2011. 136

[179] D. Ledger and D. McCaffrey. Inside Wearables: How the Science of Human Behavior Change Offers the Secret to Long-Term Engagement. Technical report, Endeavour Partners, 2014. 46, 70

[180] M. L. Lee and A. K. Dey. Real-Time Feedback for Improving Medication Taking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 2259–2268. ACM, 2014. 24

[181] Y. Lee, S. Iyengar, C. Min, Y. Ju, S. Kang, T. Park, J. Lee, Y. Rhee, and J. Song. MobiCon: A Mobile Context-Monitoring Platform. *Communications of the ACM*, 55(3):54–65, 2012. 36, 37

[182] K. Leichtenstern, E. Andre, E. Losch, M. Kranz, and P. Holleis. A Tangible User Interface as Interaction and Presentation Device to a Social Learning Software. In *Proceedings of the 4th International Conference on Networked Sensing Systems (INSS)*, pages 114–117, 2007. 29, 71

[183] D. Lemire and A. Maclachlan. Slope One Predictors for Online Rating-Based Collaborative Filtering. In *Proceedings of the SIAM Data Mining Conference (SDM)*, pages 1–5, 2005. 79

[184] S. Lemmelä, A. Vetek, K. Mäkelä, and D. Trendafilov. Designing and Evaluating Multimodal Interaction for Mobile Contexts. In *Proceedings of the 10th International Conference on Multimodal Interfaces (ICMI)*, pages 265–272. ACM, 2008. 19, 20, 22, 23

[185] J. Lester, T. Choudhury, and G. Borriello. A Practical Approach to Recognizing Physical Activities. In *Proceedings of the 4th International Conference on Pervasive Computing (Pervasive)*, pages 1–16. Springer, 2006. 25, 41, 70

[186] M. S. Lew, N. Sebe, C. Djeraba, and R. Jain. Content-Based Multimedia Information Retrieval: State of the Art and Challenges. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 2(1):1–19, 2006. 9

[187] B. Li, J. Salter, A. Dempster, and C. Rizos. Indoor Positioning Techniques Based on Wireless LAN. In *Proceedings of the 1st IEEE International Conference on Wireless Broadband and Ultra Wideband Communications*, pages 13–16. IEEE, 2006. 33

[188] G. Light, S. Calkins, and R. Cox. *Learning and Teaching in Higher Education: The Reflective Professional*. Sage, 2009. 75

[189] H. Lim, L. Kung, J. Hou, and H. Luo. Zero-Configuration, Robust Indoor Localization: Theory and Experimentation. Technical report, University of Illinois, 2005. 32

[190] A. Liu, H. Hile, H. Kautz, G. Borriello, P. Brown, M. Harniss, and K. Johnson. Indoor Wayfinding: Developing a Functional Interface for Individuals with Cognitive Impairments. *Disability & Rehabilitation: Assistive Technology*, 3(1-2):69–81, 2008. 35

[191] H. Liu, H. Darabi, P. Banerjee, and J. Liu. Survey of Wireless Indoor Positioning Techniques and Systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, 37(6):1067–1080, 2007. 31

[192] H. Lockheimer. Google Mobile Blog. Android and Security. `http://googlemobile.blogspot.de/2012/02/android-and-security.html`, February 2012. 178

[193] C.-K. Looi, L.-H. Wong, H.-J. So, P. Seow, Y. Toh, W. Chen, B. Zhang, C. Norris, and E. Soloway. Anatomy of a Mobilized Lesson: Learning My Way. *Computers & Education*, 53(4):1120–1132, 2009. 29

[194] A. Lorenz and R. Oppermann. Mobile Health Monitoring for the Elderly: Designing for Diversity. *Pervasive and Mobile Computing*, 5(5):478–495, 2009. 25

[195] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 32

[196] J. Luk, J. Pasquero, S. Little, K. MacLean, V. Levesque, and V. Hayward. A Role for Haptics in Mobile Interaction: Initial Design Using a Handheld Tactile Display Prototype. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 171–180. ACM, 2006. 18

[197] J. Mankoff and S. Carter. Crossing Qualitative and Quantitative Evaluation in the Domain of Ubiquitous Computing. In *Proceedings of the CHI Workshop 'Usage Analysis: Combining Logging and Qualitative Methods'*, 2005. 41

[198] H. B. Mann and D. R. Whitney. On a Test of Whether one of Two Random Variables is Stochastically Larger than the Other. *The Annals of Mathematical Statistics The Annals of Mathematical Statistics The Annals of Mathematical Statistics*, 18(1):50–60, 1947. 115

[199] W. Mann and A. Helal. Smart Phones for the Elders: Boosting the Intelligence of Smart Homes. In *Proceedings of the AAAI Workshop on Automation as Caregiver: The Role of Intelligent Technology in Elder Care*, pages 74–79, 2002. 24

[200] A. Mannini and A. Sabatini. Machine Learning Methods for Classifying Human Physical Activity from On-Body Accelerometers. *Sensors*, 10(2):1154–1175, 2010. 25

[201] M. Manuguerra and P. Petocz. Promoting Student Engagement by Integrating New Technology into Tertiary Education: The Role of the iPad. *Asian Social Science*, 7(11), 2011. 28

[202] E. Martín, R. M. Carro, and P. Rodríguez. A Mechanism to Support Context-Based Adaptation in M-Learning. In *Innovative Approaches for Learning and Knowledge Sharing*, pages 302–315. Springer, 2006. 29

[203] F. G. Martin. Will Massive Open Online Courses Change How We Teach? *Communications of the ACM*, 55(8):26–28, 2012. 28

[204] M. Martyn. The Hybrid Online Model: Good Practice. 1:18–23, 2003. 28

[205] R. L. Mason, R. F. Gunst, and J. L. Hess. *Statistical Design and Analysis of Experiments*, volume 10. John Wiley & Sons, 2003. 55

[206] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust Wide-Baseline Stereo from Maximally Stable Extremal Regions. *Image and Vision Computing*, 22(10):761–767, 2004. 32, 48

[207] R. Mautz. Overview of Current Indoor Positioning Systems. *Geodezija ir kartografija*, 35(1):18–22, 2009. 31

[208] R. McCarney, J. Warner, S. Iliffe, R. van Haselen, M. Griffin, and P. Fisher. The Hawthorne Effect: A Randomised, Controlled Trial. *BMC Medical Research Methodology*, 7(1):30, 2007. 41

[209] D. McMillan, A. Morrison, O. Brown, M. Hall, and M. Chalmers. Further Into the Wild: Running Worldwide Trials of Mobile Systems. In *Proceedings of the 8th International Conference on Pervasive Computing (Pervasive)*, pages 210–227. Springer, 2010. 175, 176

[210] T. Meilinger, C. Hölscher, S. J. Büchner, and M. Brösamle. How Much Information Do You Need? Schematic Maps in Wayfinding and Self Localisation. In *Proceedings of the International Conference on Spatial Cognition (SC): Reasoning, Action, Interaction*, pages 381–400. Springer, 2007. 34

[211] B. Meyer, A. Möller, A. Thielsch, A. Hendrich, and M. Kranz. Förderung der Methodenkompetenz von Lehrenden an Hochschulen – Design-Based Research rund um 'MobiDics'. In *77. Tagung der AEPF (Arbeitsgruppe für Empirische Pädagogische Forschung)*, pages 223–223, 2012. 72

[212] F. Michahelles and B. Schiele. Sensing and Monitoring Professional Skiers. *IEEE Pervasive Computing*, pages 40–46, 2005. 25

[213] G. Milette and A. Stroud. *Professional Android Sensor Programming*. John Wiley & Sons, 2012. 13

[214] A. Millonig and K. Schechtner. Developing Landmark-Based Pedestrian-Navigation Systems. *IEEE Transaction on Intelligent Transportation Systems*, 8(1):43–49, 2007. 34

[215] E. Miluzzo, N. Lane, H. Lu, and A. Campbell. Research in the App Store Era: Experiences from the CenceMe App Deployment on the iPhone. In *Proceedings of the 1st International Workshop on Research in the Large. Held in Conjunction with UbiComp*, 2010. 42, 176

[216] F. Miranda, T. Cabral Ferreira, J. Pimentão, and P. Sousa. Review on Context Classification in Robotics. In M. Kryszkiewicz, C. Cornelis, D. Ciucci, J. Medina-Moreno, H. Motoda, and Z. Raś, editors, *Rough Sets and Intelligent Systems Paradigms*, volume 8537 of *Lecture Notes in Computer Science*, pages 269–276. Springer International Publishing, 2014. 127

[217] T. Miyashita, P. Meier, T. Tachikawa, S. Orlic, T. Eble, V. Scholz, A. Gapel, O. Gerl, S. Arnaudov, and S. Lieberknecht. An Augmented Reality Museum Guide. In *Proceedings of the 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pages 103–106. IEEE, 2008. 35

[218] Y. Miyazaki and T. Kamiya. Pedestrian Navigation System for Mobile Phones Using Panoramic Landscape Images. In *Proceedings of the International Symposium on Applications and the Internet (SAINT)*. IEEE, 2006. 35

[219] A. Möller, B. Beege, S. Diewald, L. Roalter, and M. Kranz. MobiDics – Cooperative Mobile E-Learning for Teachers. In M. Specht, J. Multisilta, and M. Sharples, editors, *Proceedings of the 11th World Conference on Mobile and Contextual Learning (mLearn)*, pages 109–116, 2012. 72

[220] A. Möller, S. Diewald, and M. Kranz. Will I Tell You Where I Am? How Spatially Limited Social Networks Affect Location Sharing Behavior. Technical report, Technische Universität München, 2014. 72, 81

[221] 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 (NordiCHI)*, pages 31–40. ACM, 2012. 20, 39, 46

[222] A. Möller, S. Diewald, L. Roalter, and M. Kranz. M3I: A Framework for Mobile Multimodal Interaction. In *Proceedings of Mensch & Computer 2014: Interaktiv unterwegs – Freiräume gestalten*, pages 355–358. Oldenbourg Verlag, 2014. 123

[223] A. Möller, S. Diewald, L. Roalter, and M. Kranz. Supporting Mobile Multimodal Interaction with a Rule-Based Framework. Technical report, Technische Universität München, 2014. 123

[224] A. Möller, S. Diewald, L. Roalter, T. Stockinger, R. Huitl, S. Hilsenbeck, and M. Kranz. Navigating Indoors Using Decision Points. In *Computer Aided Systems Theory - EUROCAST 2013*, volume 8112 of *Lecture Notes in Computer Science*, pages 450–457. Springer Berlin Heidelberg, Feb. 2013. 86, 160

[225] A. Möller, M. Kranz, S. Diewald, L. Roalter, R. Huitl, T. Stockinger, M. Koelle, and P. Lindemann. Experimental Evaluation of User Interfaces for Visual Indoor Navigation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*. ACM, 2014. 78, 82, 86, 160

[226] A. Möller, M. Kranz, R. Huitl, S. Diewald, and L. Roalter. A Mobile Indoor Navigation System Interface Adapted to Vision-Based Localization. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia (MUM)*, pages 4:1–4:10. ACM, 2012. 39, 86

[227] A. Möller, M. Kranz, L. Roalter, and S. Diewald. Decision-Point Panorama-Based Indoor Navigation. In *Proceedings of the 14th International Conference on Computer Aided Systems Theory (EUROCAST)*, pages 325–326, 2013. 86

[228] 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 (CHI)*, pages 2931–2940. ACM, 2013. 81, 157

[229] 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*, 2012. 86

[230] A. Möller, F. Michahelles, S. Diewald, L. Roalter, and M. Kranz. Update Behavior in App Markets and Security Implications: A Case Study in Google Play. In *Proceedings of the 3rd International Workshop on Research in the Large. Held in Conjunction with Mobile HCI*, pages 3–6, 2012. 82, 157

[231] A. Möller, L. Roalter, S. Diewald, J. Scherr, M. Kranz, N. Hammerla, P. Olivier, and T. Plötz. GymSkill: A Personal Trainer for Physical Exercises. In *Proceedings of the IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 213–220. IEEE, 2012. 46, 59, 62, 65

[232] A. Möller, L. Roalter, and M. Kranz. Cognitive Objects for Human-Computer Interaction and Human-Robot Interaction. In *Proceedings of the 6th International Conference on Human-Robot Interaction (HRI)*, pages 207–208. ACM, 2011. 46

[233] A. Möller, J. Scherr, L. Roalter, S. Diewald, N. Hammerla, T. Plötz, P. Olivier, and M. Kranz. GymSkill: Mobile Exercise Skill Assessment to Support Personal Health and Fitness. In *Video Proceedings of the 9th International Conference on Pervasive Computing (Pervasive)*, 2011. 46

[234] A. Möller, A. Thielsch, B. Dallmeier, A. Hendrich, B. E. Meyer, L. Roalter, S. Diewald, and M. Kranz. MobiDics – Eine mobile Didaktik-Toolbox für die universitäre Lehre. In H. Rohland, A. Kienle, and S. Friedrich, editors, *DeLFI 2011 - Die 9. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V.*, volume 188 of *LNI*, pages 139–150. GI, 2011. 72

[235] A. Möller, A. Thielsch, B. Dallmeier, L. Roalter, S. Diewald, A. Hendrich, B. E. Meyer, and M. Kranz. MobiDics – Improving University Education With A Mobile Didactics Toolbox. In *Video Proceedings of the 9th International Conference on Pervasive Computing (Pervasive)*, 2011. 72

[236] S. Moore, H. MacDougall, J. Gracies, H. Cohen, and W. Ondo. Long-Term Monitoring of Gait in Parkinson's Disease. *Gait & Posture*, 26(2):200–207, 2007. 25

[237] R. Moreno and R. Mayer. Interactive Multimodal Learning Environments. *Educational Psychology Review*, 19(3):309–326, 2007. 29

[238] A. Mulloni, H. Seichter, A. Dünser, P. Baudisch, and D. Schmalstieg. 360° Panoramic Overviews for Location-Based Services. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 2565–2568. ACM, 2012. 35

[239] A. Mulloni, D. Wagner, I. Barakonyi, and D. Schmalstieg. Indoor Positioning and Navigation with Camera Phones. *IEEE Pervasive Computing*, 8(2):22–31, 2009. 31, 32, 33

[240] E. D. Mynatt, J. Rowan, S. Craighill, and A. Jacobs. Digital Family Portraits: Supporting Peace of Mind for Extended Family Members. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 333–340. ACM, 2001. 24

[241] L. Naismith, P. Lonsdale, G. Vavoula, and M. Sharples. Literature Review in Mobile Technologies and Learning. *FutureLab Report*, 11, 2004. 28

[242] W. Narzt, G. Pomberger, A. Ferscha, D. Kolb, R. Müller, J. Wieghardt, H. Hörtner, and C. Lindinger. Augmented Reality Navigation Systems. *Universal Access in the Information Society*, 4(3):177–187, 2006. 35

[243] J. Nielsen. *Usability Engineering*. Morgan Kaufmann, 1993. 41

[244] J. Nielsen. Heuristic Evaluation. In J. Nielsen and R. L. Mack, editors, *Usability Inspection Methods*. John Wiley & Sons, 1994. 26, 39

[245] J. Nielsen and T. K. Landauer. A Mathematical Model of the Finding of Usability Problems. In *Proceedings of the INTERACT and CHI Conference on Human Factors in Computing Systems*, pages 206–213. ACM, 1993. 127

[246] L. Nigay and J. Coutaz. A Design Space for Multimodal Systems: Concurrent Processing and Data Fusion. In *Proceedings of the INTERACT and CHI Conference on Human Factors in Computing Systems*, pages 172–178. ACM, 1993. 9, 10, 21

[247] D. A. Norman. *The Psychology of Everyday Things*. Basic books, 1988. 91

[248] D. A. Norman. The 'Problem' with Automation: Inappropriate Feedback and Interaction, not 'Over-Automation'. *Philosophical Transactions of the Royal Society of London. B, Biological Sciences*, 327(1241):585–593, 1990. 22

[249] D. A. Norman. Emotion & Design: Attractive Things Work Better. *ACM interactions*, 9(4):36–42, 2002. 19, 34

[250] C. Obolashvili. *Evaluating Socially Connected E-Learning*. Master thesis, Technische Universität München, Germany, 2013. 80, 81

[251] H. Ogata, M. Li, B. Hou, N. Uosaki, M. M. El-Bishouty, and Y. Yano. Ubiquitous Learning Log: What If We Can Log Our Ubiquitous Learning. In *Proceedings of the 18th International Conference on Computers in Education*, pages 360–367, 2010. 29

[252] L. R. Olsen, P. Chaudhuri, and F. Godtliebsen. Multiscale Spectral Analysis for Detecting Short and Long Range Change Points in Time Series. *Computational Statistics & Data Analysis*, 52:3310–3330, 2008. 64

[253] E. O'Neill, P. Thompson, S. Garzonis, and A. Warr. Reach Out and Touch: Using NFC and 2D Barcodes for Service Discovery and Interaction with Mobile Devices. *Proceedings of the 5th International Conference on Pervasive Computing (Pervasive)*, pages 19–36, 2007. 48

[254] S. Oviatt. Multimodal Interactive Maps: Designing for Human Performance. *Human-Computer Interaction*, 12(1):93–129, 1997. 21, 22, 23

[255] S. Oviatt. Ten Myths Of Multimodal Interaction. *Communications of the ACM*, 42(11):74–81, 1999. 3, 21, 23

[256] P. Palanque and R. Bastide. Petri Net Based Design of User-Driven Interfaces Using the Interactive Cooperative Objects Formalism. In F. Paternó, editor, *Interactive Systems: Design, Specification, and Verification*, Focus on Computer Graphics, pages 383–400. Springer Berlin Heidelberg, 1995. 38

[257] L. Palen and M. Salzman. Voice-Mail Diary Studies for Naturalistic Data Capture Under Mobile Conditions. In *Proceedings of the ACM Conference on Computer Supported Cooperative Work (CSCW)*, pages 87–95. ACM, 2002. 41, 42

[258] G. Paolacci, J. Chandler, and P. G. Ipeirotis. Running Experiments on Amazon Mechanical Turk. *Judgment and Decision Making*, 5(5):411–419, 2010. 174

[259] A. Parker and J. Tritter. Focus Group Method and Methodology: Current Practice and Recent Debate. *International Journal of Research & Method in Education*, 29(1):23–37, 2006. 39

[260] M. J. Pazzani and D. Billsus. Content-Based Recommendation Systems. In *The Adaptive Web*, pages 325–341. Springer, 2007. 79

[261] K. Pearson. Note on Regression and Inheritance in the Case of Two Parents. *Proceedings of the Royal Society of London*, 58(347-352):240–242, 1895. 69

[262] C. Pham-Nguyen and S. Garlatti. Context-Aware Scenarios for Pervasive Long-Life Learning. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology (WI-IAT)*, pages 824–827. IEEE, 2008. 29

[263] R. E. Ployhart and R. J. Vandenberg. Longitudinal Research: The Theory, Design, and Analysis of Change. *Journal of Management*, 36(1):94–120, 2009. 40

[264] M. Prasad, P. Taele, D. Goldberg, and T. A. Hammond. HaptiMoto: Turn-By-Turn Haptic Route Guidance Interface for Motorcyclists. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 3597–3606. ACM, 2014. 35

[265] M. Prasad, P. Taele, A. Olubeko, and T. A. Hammond. HaptiGo: A Navigational 'Tap on the Shoulder'. In *IEEE Haptics Symposium (HAPTICS)*, pages 1–7. IEEE, 2014. 35

[266] 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)*, 9(3):171–193, 2002. 21, 22

[267] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng. ROS: An Open-Source Robot Operating System. In *ICRA Workshop on Open Source Software*, volume 3, page 5, 2009. 136

[268] M. Raento, A. Oulasvirta, R. Petit, and H. Toivonen. ContextPhone: A Prototyping Platform for Context-Aware Mobile Applications. *IEEE Pervasive Computing*, 4(2):51–59, 2005. 36

[269] P. Ramsden. *Learning to Teach in Higher Education*. Routledge, London/New York, 1992. 29, 75

[270] A. Ratzka. Steps in Identifying Interaction Design Patterns for Multimodal Systems. In P. Forbrig and F. Paternò, editors, *Engineering Interactive Systems*, volume 5247 of *Lecture Notes in Computer Science*, pages 58–71. Springer Berlin Heidelberg, 2008. 18, 20

[271] A. Ratzka. User Interface Patterns for Multimodal Interaction. In J. Noble, R. Johnson, U. Zdun, and E. Wallingford, editors, *Transactions on Pattern Languages of Programming III*, volume 7840 of *Lecture Notes in Computer Science*, pages 111–167. Springer Berlin Heidelberg, 2013. 18, 19, 20

[272] M. Raubal and S. Winter. Enriching Wayfinding Instructions with Local Landmarks. In M. Egenhofer and D. Mark, editors, *Geographic Information Science*, volume 2478 of *Lecture Notes in Computer Science*, pages 243–259. Springer Berlin Heidelberg, 2002. 109

[273] L. Ravindranath, A. Thiagarajan, H. Balakrishnan, and S. Madden. Code in the Air: Simplifying Sensing and Coordination Tasks on Smartphones. In *Proceedings of the 12th Workshop on Mobile Computing Systems & Applications*, page 4. ACM, 2012. 38, 127

[274] L. M. Reeves, J. Lai, J. A. Larson, S. Oviatt, T. S. Balaji, S. Buisine, P. Collings, P. Cohen, B. Kraal, J.-C. Martin, M. McTear, T. Raman, K. M. Stanney, H. Su, and Q. Y. Wang. Guidelines for Multimodal User Interface Design. *Communications of the ACM*, 47(1):57–59, 2004. 19

[275] 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 (CHI)*, pages 887–896. ACM, 2010. 21, 23, 160

[276] P. Riedl, P. Koller, R. Mayrhofer, M. Kranz, A. Möller, and M. Koelle. Visualizations and Switching Mechanisms for Security Zones. In *Proceedings of the 11th International Conference on Advances in Mobile Computing and Multimedia (MoMM)*, 2013. 181

[277] L. Roalter, M. Kranz, S. Diewald, and A. Möller. The Smartphone as Mobile Authorization Proxy. In *Proceedings of the 14th International Conference on Computer Aided Systems Theory (EUROCAST 2013)*, pages 306–307, 2013. 81

[278] L. Roalter, M. Kranz, S. Diewald, and A. Möller. User-Friendly Authentication and Authorization Using a Smartphone Proxy. In *Computer Aided Systems Theory - EUROCAST 2013*, volume 8112 of *Lecture Notes in Computer Science*, pages 390–399. Springer Berlin Heidelberg, Feb. 2013. 81, 82

[279] L. Roalter, M. Kranz, and A. Möller. A Middleware for Intelligent Environments and the Internet of Things. In Z. Yu, R. Liscano, G. Chen, D. Zhang, and X. Zhou, editors, *Ubiquitous Intelligence and Computing*, volume 6406 of *Lecture Notes in Computer Science*, pages 267–281. Springer Berlin / Heidelberg, 2010. 30

[280] L. Roalter, M. Kranz, A. Möller, S. Diewald, T. Stockinger, M. Koelle, and P. Lindemann. Visual Authentication: A Secure Single Step Authentication for User Authorization. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia (MUM)*, pages 30:1–30:4. ACM, 2013. 30, 71, 81, 83

[281] L. Roalter, T. Linner, A. Möller, S. Diewald, and M. Kranz. The Healthcare and Motivation Seat – A Survey with the GewoS Chair. In *Mensch & Computer Workshopband*, pages 37–43, 2012. 68

[282] L. Roalter, A. Möller, S. Diewald, and M. Kranz. Developing Intelligent Environments: A Development Tool Chain for Creation, Testing and Simulation of Smart and Intelligent Environments. In *Proceedings of the 7th International Conference on Intelligent Environments (IE)*, pages 214–221, 2011. 30, 71

[283] M. Röckl, T. Strang, and M. Kranz. V2V Communications in Automotive Multi-Sensor Multi-Target Tracking. In *Proceedings of the IEEE Vehicular Technology Conference (VTC)*, pages 1–5, 2008. 119

[284] F. J. Roethlisberger and W. J. Dickson. *Management and the Worker: An Account of a Research Program Conducted by the Western Electric Company, Hawthorne Works, Chicago*. Harvard University Press, 1964. 172

[285] T. Ross, A. May, and S. Thompson. The Use of Landmarks in Pedestrian Navigation Instructions and the Effects of Context. In *Proceedings of the 6th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI)*, pages 300–304. Springer, 2004. 34

[286] E. Rosten and T. Drummond. Machine Learning for High-Speed Corner Detection. In *Proceedings of the 9th European Conference on Computer Vision (ECCV)*, pages 430–443. Springer, 2006. 32, 99

[287] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An Efficient Alternative to SIFT or SURF. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2564–2571. IEEE, 2011. 32

[288] T. Rudd. Learning Spaces and Personalisation Workshop Outcomes. *NESTA Futurelab*, 16, 2008. 28

[289] 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 (CHI)*, pages 197–206. ACM, 2011. 21

[290] E. Rukzio, K. Leichtenstern, V. Callaghan, P. Holleis, A. Schmidt, and J. Chin. An Experimental Comparison of Physical Mobile Interaction Techniques: Touching, Pointing and Scanning. *Proceedings of the 8th International Conference on Ubiquitous Computing (UbiComp)*, pages 87–104, 2006. 39, 46, 47, 48

[291] F. Sadri. Ambient Intelligence: A Survey. *ACM Computing Surveys (CSUR)*, 43(4):36, 2011. 10

[292] A. Sahami, N. Henze, T. Dingler, K. Kunze, and A. Schmidt. Upright or Sideways?: Analysis of Smartphone Postures in the Wild. In *Proceedings of the 15th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI)*, pages 362–371. ACM, 2013. 42, 175, 176

[293] A. Sahami, N. Henze, T. Dingler, M. Pielot, D. Weber, and A. Schmidt. Large-Scale Assessment of Mobile Notifications. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 3055–3064. ACM, 2014. 42, 175, 176

[294] M. Sahmoudi, M. G. Amin, and R. Landry. Acquisition of Weak GNSS Signals Using a New Block Averaging Pre-Processing. In *Proceedings of the IEEE/ION Position, Location and Navigation Symposium*, pages 1362–1372. IEEE, 2008. 31

[295] M. Sauter. *Grundkurs Mobile Kommunikationssysteme: UMTS, HSDPA und LTE, GSM, GPRS und Wireless LAN*. Springer, 2010. 15

[296] J. B. Schafer, D. Frankowski, J. Herlocker, and S. Sen. Collaborative Filtering Recommender Systems. In *The Adaptive Web*, pages 291–324. Springer, 2007. 79

[297] 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 (HUC)*, pages 89–101. Springer, 1999. 134

[298] A. Schmidt, P. Holleis, and M. Kranz. Sensor-Virrig – A Balance Cushion as Controller. *Proceedings of the UbiComp Workshop 'Playing with Sensors'*, 2004. 25, 29

[299] A. Schmidt, M. Kranz, and P. Holleis. Embedded Information. *Proceedings of the UbiComp Workshop 'Ubiquitous Display Environments'*, 2004. 30

[300] A. Schmidt, M. Kranz, and P. Holleis. Interacting with the Ubiquitous Computer: Towards Embedding Interaction. In *Proceedings of the Joint Conference on Smart Objects and Ambient Intelligence (sOc-EUSAI)*, pages 147–152. ACM, 2005. 4

[301] G. Schroth, S. Hilsenbeck, R. Huitl, F. Schweiger, and E. Steinbach. Exploiting Text-Related Features for Content-Based Image Retrieval. In *IEEE International Symposium on Multimedia (ISM)*, pages 77–84. IEEE, 2011. 49

[302] G. Schroth, R. Huitl, D. Chen, M. Abu-Alqumsan, A. Al-Nuaimi, and E. Steinbach. Mobile Visual Location Recognition. *IEEE Signal Processing Magazine*, 28(4):77–89, 2011. 32, 33, 118

[303] C. Schuster, M. Appeltauer, and R. Hirschfeld. Context-Oriented Programming for Mobile Devices: JCop on Android. In *Proceedings of the 3rd International Workshop on Context-Oriented Programming*, page 5. ACM, 2011. 37

[304] P. Seppälä and H. Alamäki. Mobile Learning in Teacher Training. *Journal of Computer Assisted Learning*, 19(3):330–335, 2003. 29

[305] W. R. Shadish, T. D. Cook, and D. T. Campbell. *Experimental and Quasi-Experimental Designs for Generalized Causal Inference*. Wadsworth Cengage Learning, 2002. 177

[306] T. C. Shan and W. W. Hua. Taxonomy of Java Web Application Frameworks. In *IEEE International Conference on e-Business Engineering (ICEBE)*, pages 378–385. IEEE, 2006. 36

[307] M. Sharples. The Design of Personal Mobile Technologies for Lifelong Learning. *Computer Education*, 34(3-4):177–193, 2000. 28, 29

[308] B. Shneiderman. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, volume 2. Addison-Wesley Reading, MA, 1992. 3, 23, 25, 127, 141

[309] V. Simpson and M. Oliver. Electronic Voting Systems for Lectures Then and Now: A Comparison of Research and Practice. *Australasian Journal of Educational Technology*, 23(2):187, 2007. 30

[310] F. Soualah-Alila, F. Mendes, and C. Nicolle. A Context-Based Adaptation In Mobile Learning. *IEEE Computer Society Technical Committee on Learning Technology (TCLT)*, 15(4), 2013. 29

[311] A. Stanciulescu, Q. Limbourg, J. Vanderdonckt, B. Michotte, and F. Montero. A Transformational Approach for Multimodal Web User Interfaces based on UsiXML. In *Proceedings of the 7th International Conference on Multimodal Interfaces (ICMI)*, pages 259–266. ACM, 2005. 36

[312] K. Stawarz, A. L. Cox, and A. Blandford. Don't Forget Your Pill!: Designing Effective Medication Reminder Apps That Support Users' Daily Routines. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 2269–2278. ACM, 2014. 24

[313] J. Steuer. Defining Virtual Reality: Dimensions Determining Telepresence. *Journal of Communication*, 42(4):73–93, 1992. 87

[314] T. Stockinger, M. Koelle, P. Lindemann, M. Kranz, S. Diewald, A. Möller, and L. Roalter. Towards Leveraging Behavioral Economics in Mobile Application Design. In T. Reiners and L. C. Wood, editors, *Gamification in Education and Business*. Springer, 2014. 46

[315] T. Stockinger, M. Koelle, P. Lindemann, L. Witzani, and M. Kranz. SmartPiggy: A Piggy Bank That Talks to Your Smartphone. In *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia (MUM)*, pages 42:1–42:2. ACM, 2013. 46

[316] J. Straub, S. Hilsenbeck, G. Schroth, R. Huitl, A. Möller, and E. Steinbach. Fast Relocalization for Visual Odometry Using Binary Features. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2013. 107

[317] C. Strohrmann, H. Harms, G. Tröster, S. Hensler, and R. Müller. Out of the Lab and Into the Woods: Kinematic Analysis in Running Using Wearable Sensors. In *Proceedings of the 13th International Conference on Ubiquitous Computing (UbiComp)*, pages 119–122. ACM, 2011. 26

[318] Student. The Probable Error of a Mean. *Biometrika*, 6(1):1–25, 1908. 98

[319] K. Synnes, M. Kranz, J. Rana, O. Schélen, and M. Nilsson. User-Centric Social Interaction for Digital Cities. In B. Guo, D. Riboni, and P. Hu, editors, *Creating Personal, Social, and Urban Awareness through Pervasive Computing*. IGI Global, Oct. 2013. 22

[320] D. Tatar, J. Roschelle, P. Vahey, and W. R. Penuel. Handhelds Go to School: Lessons Learned. *Computer*, 36(9):30–37, Sept. 2003. 28, 29

[321] A. S. Taylor. Ethnography in Ubiquitous Computing. In J. Krumm, editor, *Ubiquitous Computing Fundamentals*. Chapman & Hall / CRC, 2010. 40

[322] L. Terrenghi, M. Kranz, P. Holleis, and A. Schmidt. A Cube to Learn: A Tangible User Interface for the Design of a Learning Appliance. *Personal and Ubiquitous Computing*, 10(2-3):153–158, 2006. 29

[323] H. Thüs, M. A. Chatti, E. Yalcin, C. Pallasch, B. Kyryliuk, T. Mageramov, and U. Schroeder. Mobile Learning in Context. *International Journal of Technology Enhanced Learning*, 4(5):332–344, 2012. 29

[324] M. Uyttendaele, A. Eden, and R. Skeliski. Eliminating Ghosting and Exposure Artifacts in Image Mosaics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages II–509. IEEE, 2001. 108

[325] P. Välkkynen, I. Korhonen, J. Plomp, T. Tuomisto, L. Cluitmans, H. Ailisto, and H. Seppä. A User Interaction Paradigm for Physical Browsing and Near-Object Control based on Tags. In *Proceedings of the Physical Interaction Workshop on Real World User Interfaces on Mobile HCI*, pages 31–34, 2003. 46, 47, 48, 52

[326] A. Van Dam. Post-WIMP User Interfaces. *Communications of the ACM*, 40(2):63–67, 1997. 77

[327] M. van den Akker, F. Buntinx, J. F. Metsemakers, S. Roos, and J. A. Knottnerus. Multimorbidity in General Practice: Prevalence, Incidence, and Determinants Of Co-Occurring Chronic and Recurrent Diseases. *Journal of Clinical Epidemiology*, 51(5):367–375, 1998. 24, 45

[328] M. W. Van Someren, Y. F. Barnard, and J. A. Sandberg. *The Think Aloud Method: A*

Practical Guide to Modelling Cognitive Processes. Academic Press London, 1994. 54, 101, 183

[329] D. Vanacken, J. De Boeck, C. Raymaekers, and K. Coninx. NiMMiT: A Notation for Modeling Multimodal Interaction Techniques. In *Proceedings of the 1st International Conference on Computer Graphics Theory and Applications (GRAPP)*, pages 224–231, 2006. 36, 38

[330] M. Van't Hooft, K. Swan, D. Cook, and Y. Lin. What Is Ubiquitous Computing? In M. Van't Hooft and K. Swan, editors, *Ubiquitous Computing in Education: Invisible Technology, Visible Impact*. Lawrence Erlbaum Associates, 2007. 30

[331] K. Venkataraman, D. Lelescu, J. Duparré, A. McMahon, G. Molina, P. Chatterjee, R. Mullis, and S. Nayar. PiCam: An Ultra-Thin High Performance Monolithic Camera Array. *ACM Transactions on Graphics (TOG)*, 32(6):166, 2013. 13

[332] J. Ventura and T. Hollerer. Wide-Area Scene Mapping for Mobile Visual Tracking. In *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 3–12. IEEE, 2012. 90

[333] E. Vodvarsky, P. Holleis, M. Kranz, and A. Schmidt. Mobile Platforms for Playful Learning and Interaction. In *Proceedings of the IADIS International Conference on Mobile Learning (ML)*, pages 214–217. IADIS Press, 2007. ISBN 978-972-8924-36-2. 29, 71

[334] B. Walther-Franks and R. Malaka. Evaluation of an Augmented Photograph-Based Pedestrian Navigation System. In A. Butz, B. Fisher, A. Krüger, P. Olivier, and M. Christie, editors, *Smart Graphics*, volume 5166 of *Lecture Notes in Computer Science*, pages 94–105. Springer Berlin Heidelberg, 2008. 35

[335] D. Wang, R. Hao, and D. Lee. Fault Detection in Rule-Based Software Systems. *Information and Software Technology*, 45(12):865–871, 2003. 137

[336] G. Wang. Designing Smule's iPhone Ocarina. In *Proceedings of the International Conference on New Interfaces for Musical Expression (NIME)*, 2009. 176

[337] Y.-K. Wang. Context Awareness and Adaptation in Mobile Learning. In *Proceedings of the 2nd IEEE International Workshop on Wireless and Mobile Technologies in Education*, pages 154–158, 2004. 29

[338] R. Want, K. Fishkin, A. Gujar, and B. Harrison. Bridging Physical and Virtual Worlds with Electronic Tags. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 370–377. ACM, 1999. 47

[339] R. Wasinger, A. Krüger, and O. Jacobs. Integrating Intra and Extra Gestures Into a Mobile and Multimodal Shopping Assistant. In *Proceedings of the 3rd International Conference on Pervasive Computing (Pervasive)*, pages 297–314. Springer, 2005. 17, 18

[340] K. Watanuki, K. Sakamoto, and F. Togawa. Multimodal Interaction in Human Communication. *IEICE TRANSACTIONS on Information and Systems*, 78(6):609–615, 1995. 21

[341] M. Weigel, V. Mehta, and J. Steimle. More Than Touch: Understanding How People Use

Skin as an Input Surface for Mobile Computing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI)*, pages 179–188. ACM, 2014. 17

[342] M. Weiser. The Computer for the 21st Century. *Scientific American*, 265(3):94–104, 1991. 189

[343] M. Werner, M. Kessel, and C. Marouane. Indoor Positioning Using Smartphone Camera. In *Proceedings of the International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–6. IEEE, 2011. 33

[344] B. Wheeler and S. Waggener. Above-Campus Services: Shaping the Promise of Cloud Computing for Higher Education. *Educause Review*, 44(6):52–67, 2009. 30

[345] F. Wilcoxon. Individual Comparisons by Ranking Methods. *Biometrics Bulletin*, 1(6):80–83, 1945. 93

[346] A. Winteler. Lehrende an Hochschulen. In A. Krapp and B. Weidenmann, editors, *Lehrbuch Pädagogische Psychologie*, pages 332–346. Psychologie Verlags Union, 2001. 29

[347] F. Wolff, A. De Angeli, and L. Romary. Acting on a Visual World: The Role of Perception in Multimodal HCI. In *Proceedings of the AAAI Workshop 'Representations for Multimodal Human-Computer Interaction'*, page 6, 1998. 18, 23

[348] O. Woodman and R. Harle. Pedestrian Localisation for Indoor Environments. In *Proceedings of the 10th International Conference on Ubiquitous Computing (UbiComp)*, pages 114–123. ACM, 2008. 31

[349] R. K. Yin. *Case Study Research: Design and Methods*. Sage Publications, 5th edition, 2009. 40

[350] S. Zhai, P. O. Kristensson, P. Gong, M. Greiner, S. A. Peng, L. M. Liu, and A. Dunnigan. Shapewriter on the iPhone: From the Laboratory to the Real World. In *Extended Abstracts on Human Factors in Computing Systems (CHI)*, pages 2667–2670. ACM, 2009. 176

[351] X. Zhang, L. Xu, L. Xu, and D. Xu. Direction of Departure (DOD) and Direction of Arrival (DOA) Estimation in MIMO Radar with Reduced-Dimension MUSIC. *IEEE Communications Letters*, 14(12):1161–1163, 2010. 32