Looking Inside

Mobile Screen Recordings as a Privacy Friendly Long-Term Data Source to Analyze User Behavior

Dissertation by Philipp Krieter

Dissertation by Philipp Krieter submitted for the degree of Doctor of Engineering (Dr.-Ing.) to the faculty of Mathematics and Computer Science at the University of Bremen.

Bremen, May 8, 2020

Supervisors
Prof. Dr. Andreas Breiter, University of Bremen
Prof. Dr. Barry Brown, Stockholm University

Acknowledgements

First of all, my special thanks go to my supervisor Andreas Breiter for his constant trust in me and my work as well as for intensive and motivating discussions and his comprehensive advice and support. I could not imagine better supervision and conditions as at the ifib to do a Ph.D.

I would also like to especially thank Barry Brown, for joining my Ph.D. project as a supervisor and for the stimulating discussions and constructive advice and feedback I received during my doctorate.

I am very grateful for all the great and amazing colleagues at the ifib who made my Ph.D. time great: thank you! Especially I want to thank Hendrik Heuer for countless helpful discussions and reading my papers and thesis and of course all the other Ph.D. students for their support and feedback. I would also like to thank my great and supporting colleagues from the musicalytics project, Julia Finken, Benjamin Weyel, and Andreas Lehmann-Wermser. I would like to thank Juliane Jarke, who came up with the idea of doing my Ph.D. at the ifib after finishing my master thesis.

Also, I would like to thank Luca and Naiana. And I want to thank Olga for supporting me during the creation of this dissertation. Finally, I would like to thank my family and all the other kind people who have supported me over the last three years.

Abstract

Mobile devices are ubiquitous in many societies and shape the way we interact with technology and each other. Research on how we use and perceive technology is essential to understand its impact. This work advances how we can follow user behavior on mobile devices. We combine the strength of two common data sources for tracking on mobile devices, log files, and screen recordings. Log files are suitable for long-term and privacy-friendly analyzation but provide rather general data (e.g. system log files) unless one has access to the source code of the applications or operating systems. Screen recordings are usually used for short-termed analysis (e.g. usability tests) because the analysis is time-consuming, but they provide all activities on the screen in high detail regardless of which application or operating system. This thesis combines both data sources and presents an approach to automatically generate log files from mobile screen recordings. The approach utilizes methods of computer vision and machine learning to automatically process screen recordings and extend their use. Screen recordings reveal virtually everything a user does with a device, making privacy important, especially in user studies. We present a privacy concept and implementation and show how the risk of exposing private data can be reduced, by processing all recordings locally on the mobile devices and anonymizing the resulting log files. In order to apply the developed method in practice, we carry out a study in the context of education and show how log files of screen recordings can complement and extend existing research in learning analytics. This thesis opens up novel perspectives on how we can look at human-computer interaction with mobile devices. We show how to generate long-term log data with high detail and accuracy from mobile screen recordings, in a privacy-friendly way, locally on mobile devices.

Zusammenfassung

Mobile Geräte sind in vielen Gesellschaften allgegenwärtig und prägen die Art und Weise, wie wir mit Technologie und untereinander interagieren. Die Erforschung der Art und Weise, wie wir Technologie nutzen und wahrnehmen, ist wesentlich, um ihre Auswirkungen zu verstehen. Diese Arbeit verbessert, wie wir das Nutzerverhalten auf mobilen Geräten verfolgen können. Wir kombinieren die Stärke von zwei gängigen Datenquellen für die Beobachtung auf mobilen Geräten, Log-Dateien und Bildschirmaufzeichnungen. Log-Dateien eignen sich für eine langfristige und datenschutzfreundliche Analyse, liefern aber eher allgemeine Daten (z.B. System-Log-Dateien), es sei denn, man hat Zugriff auf den Quellcode der Anwendungen oder Betriebssysteme. Bildschirmaufzeichnungen werden in der Regel für kurzfristige Analysen (z.B. Usability-Tests) verwendet, da die Analyse zeitaufwendig ist, sie liefern aber unabhängig von der Anwendung oder dem Betriebssystem alle Aktivitäten auf dem Bildschirm in hoher Detailtiefe. In dieser Arbeit wird ein Ansatz zur automatischen Generierung von Logdateien aus mobilen Bildschirmaufzeichnungen vorgestellt. Der Ansatz nutzt Methoden der Computer Vision und des maschinellen Lernens, um Bildschirmaufzeichnungen automatisch zu verarbeiten und deren Nutzung zu erweitern. Bildschirmaufzeichnungen offenbaren praktisch alles, was jemand mit einem Gerät macht, was den Datenschutz wichtig macht, besonders in Studien mit Nutzer*innen. Wir stellen ein Datenschutzkonzept und eine Implementierung vor und zeigen, wie das Risiko der Preisgabe privater Daten reduziert werden kann, indem alle Aufzeichnungen lokal auf den mobilen Geräten verarbeitet und die resultierenden Protokolldateien anonymisiert werden. Um die entwickelte Methode in der Praxis anzuwenden, führen wir eine Studie im Bildungskontext durch und zeigen, wie Log-Dateien von Bildschirmaufzeichnungen die bestehende Forschung in learning analytics ergänzen und erweitern können. Diese Arbeit eröffnet neue Perspektiven, wie wir die Mensch-Computer-Interaktion auf mobilen Geräten betrachten können. Wir zeigen, wie man auf datenschutzfreundliche Weise Langzeit-Logdaten mit hoher Detailtiefe und Genauigkeit aus mobilen Bildschirmaufzeichnungen lokal auf mobilen Geräten erzeugen kann.

Contents

1. Introduction

1.1. Motivation

Have you ever sat down on a train or plane and accidentally looked at the screen of your neighbor's smartphone and felt guilty because you learned something about this person that you did not intend? Just by a couple of moments of seeing someone interacting with a smartphone, we can get to know much information. From a quick view, it can be possible to infer political views from the news source someone is reading, what kind of books this person likes, or just a personal message in a private chat. The screen of someone's smartphone tells a lot about this person, which makes screen recordings an interesting data source for user behavior research and at the same time also raises obvious privacy concerns.

Mobile devices, especially smartphones and tablet computers, are ubiquitous in many societies and shape the way we interact with our environment and each other. They influence many parts of our life: relationships, health, work, education, or how we learn – just to name a few. Acquiring knowledge about the way we use and perceive technology is essential to understand its impact and this is a challenge in current research. There are thousands of research studies that aim at exploring mobile Human-Computer Interaction (HCI) in order to better understand, how we use and interact with these devices and how they influence our lives. A search for "mobile human-computer interaction" in the Scopus database results in 12,994 entries (March 19, 2020). The variety of topics in this field is very broad and affects all areas of living.

Understanding and analyzing mobile device interaction requires collecting data in the first place. Visual perception shapes human-computer interaction, which results from the design of interfaces that concentrate the strengths of the human sense of sight (Ebert et al., 2012). Most mainstream computer systems use graphical user interfaces (GUIs) to display information and to interact with users (Sears & Jacko, 2009, p.126). The focus in HCI on GUIs leads to the fact, that watching screen recordings makes it possible to understand almost any action, behavior or task a user is pursuing while using a computer

system, as a GUI always represents the current state of a system, based on user and system events (Aho, Kanstren, et al., 2014, p.55). A common use case for screen video recording and analysis is to learn how users interact with software or to evaluate the usability and user experience. Using both qualitative and quantitative methods, screen capture analysis can provide insights into how people use and interact with computer systems. It is comparable to watching over the shoulder of the user, which makes this data source highly interesting from a research perspective: we can follow everything a user is doing in high detail. But a look over the shoulder can also be an invasion of privacy.

Coming from here there are two important challenges or limitations when we want to work with screen recordings as a long-term data source. An obvious challenge is that the manual analysis of screen videos takes a significant amount of time. This makes the manual analysis of screen recordings to find certain events or actions of interest unfeasible for long-term studies involving hours, days, or months of screen recordings. This limits the analysis to rather short periods of time. Currently, most user experience evaluation methods focus on short-term evaluations (Vermeeren et al., 2010). The second major challenge is privacy: watching someone using a private mobile device can be very privacy-invasive in its nature. This further restricts the use of mobile screen recordings. We know from the desktop platform that users are skeptical when screen recordings are used over a long period of time to collect empirical data (Tang et al., 2006). Since smartphones or tablets are usually very personal devices, recording and analyzing their screen is very problematic, especially when it comes to long terms.

In contrast, for long-term analysis, log files are a common and efficient way of collecting data on mobile devices, but with different challenges. Data sources for these are mostly system logs or accessibility APIs, for example (see e.g. Ferreira et al., 2015). Existing solutions for tracking application usage on mobile devices provide information about which application has been used for how long, for example, but lack the ability to gain insight into in-app actions (Böhmer et al., 2011; Do et al., 2011). This results in log files, which can give a rather general idea of what is happening on the device. Most mobile applications are closed. This means it is not possible to just implement log commands inside a third-party application or the operating system to create custom log files. Because of that – unlike with screen recordings – it is not possible to just follow anything a user is doing inside an application or anything else on the device. As a consequence, we may, for example, know that a user is

opening and using a music application for an hour but we cannot give details about what is happening within the application. The user might learn how to play the piano or a guitar, or learn how to combine existing songs to a new one, but all we can tell is that the application was opened. We could follow what the user is doing inside the music application using a screen recording, but not just by collecting system log files. The generated data is limited to rather general system events but has, on the one hand, an advantage from a privacy perspective. The collection of anonymous usage data using system logs is relatively easy to implement. The anonymization of screen videos, on the other hand, is much more difficult.

Screen recordings can provide detailed data about everything that happens on the screen, but are limited to short periods of time and are problematic from a privacy perspective. System log files and the like are suitable for long periods of time, but they usually lack the level of detail that we can extract from screen recordings, yet they are easier to collect anonymously and therefore it is easier to protect the privacy of users. This limits the amount and level of detail of data we can collect on mobile devices, and thus the questions we can answer. The motivation and goal of this thesis is to overcome these limitations and to combine the strength of both data sources.

1.2. Education as an Application Context for Screen Recordings

In Education, mobile digital devices play an increasing role. Most students and universities use digital devices and services to organize and enable learning. At the same time, more data is being collected to learn more about how people learn in digital environments and how these processes can be optimized. The field of learning analytics deals with generating insights from this educational data with the goal of the modeling, prediction, and optimization of learning processes (more details in section 2.2). The most common data source for learning analytics are log files, which are often collected in Learning Management Systems (LMS) (Papamitsiou & Economides, 2014). An LMS is a web-based system for content management, which enables communication between students, teachers, and educational institutions. It usually provides management functions for assignments, courses, and exams. LMS are in use at most Universities and K-12 institutions. When students use an LMS

they leave digital traces, which are saved in the form of log files. They contain, for example, which pages a student has visited in the LMS, how often the student has logged into the system, or written a message to the instructor. These log files share the aspects mentioned in the beginning: they are good for long-term analysis of how students interact with an LMS, but they lack details. That is why some important measurements that are derived from these log files tent to be imprecise (more on this in section 2.2.3). This applies to the use of the LMS, as well as for things that happen outside of the LMS, which are relevant for learning. This results in the fact that the log files from an LMS can not take into account other digital learning activities on the device or communication through other existing channels (e.g. WhatsApp, Email). To give an example, a student in a music class could read an assignment in the LMS to record a song in a third-party piano application. We could trace the view of the content in the LMS, but we could not follow how the student managed to fulfill the assignment in the piano application or where problems with the assignment might have occurred because it happened outside of the LMS. This lack of detail in the log files limits the questions we can answer in this field. Detailed data of what students do in other applications related to learning is very attractive for the discipline of learning analytics to provide a bigger picture of digital learning processes. Screen recordings could give details on LMS use which the logs do not cover and additionally, what is happening beyond the LMS in other applications. For this purpose, we could manually analyze the behavior of a student based on the videos. But of course, it is not feasible to manually analyze the screen recordings of an entire semester of all students, that would simply take too much time. This makes learning analytics or respectively education an excellent field of application for the research approach presented in this dissertation.

1.3. Research Outline

This thesis is embedded in the fields of HCI and learning analytics, both inter-disciplinary parts of computer science. The field of HCI explores the interaction between humans and computers and touches a range of disciplines, computer science, social sciences and behavioral sciences, and psychology (Carroll, 1997). Contributions to this inter-disciplinary field came also from education and graphics design and aim at expanding the understanding of inter-

actions between people and technology (Sears & Jacko, 2009, p. xiii). Learning analytics combine the fields of technology, education, learning theory, and data mining with the goal of understanding and improving learning processes from data. It also connects methods from "business intelligence, web analytics, academic analytics, action analytics, and predictive analytics" (Papamitsiou & Economides, 2014). Both fields share similar methods and ways of collecting data to answer their field-specific questions. Log files are, for example, an important way to track student activities in learning analytics and also in HCI research in order to track user behavior related to mobile applications.

This thesis contributes to both fields, mobile HCI research, and learning analytics, by introducing a novel way of generating data which allows us to expand our knowledge on user behavior in HCI and digital learning environments. The idea is the combination of the strength of the short-term but high-detailed manual analysis of screen recordings with the long-term and privacy feasibility of "classic" log file analysis. We do this by utilizing computer vision and machine learning methods to automatically analyze mobile screen recordings. The main contribution is, to enable a way of collecting data using screen recordings, resulting in a highly detailed, privacy-friendly data source, suitable for long-term studies on user behavior. We enable a new perspective of how we can do research in mobile HCI and learning analytics: We make the behavior of users on mobile devices trackable from a perspective of all that is happening on the device's screen at an application-internal level of detail, regardless of which application. We implement and evaluate a privacy concept to facilitate the application of the method in user studies by reducing privacy invasion, resulting in a unique approach to privacy for screen recordings. We demonstrate how we can use the presented approach and expand the scope of questions we can answer to contribute to HCI and learning analytics. We show how to better understand and enhance common and important measurements in learning analytics, using the developed logging approach, based on long-term screen recordings.

Technically, the aim is to explore how existing methods in log file generation, computer vision, and machine learning can be used to automatically analyze screen recordings to track user interactions on mobile devices, independent of which or how many different applications or having access to their source code. The idea is to collect data with the information density and details of the analysis of a mobile screen recording by creating log files that can provide as rich and in-depth information as watching a user interact with a

screen but without having to watch hours or days of recorded video material. The assumption is, that the presented method of log file generation results in a powerful tool applicable in multiple domains like for analysis of mobile HCI, learning analytics, software testing, requirement engineering, and other domains. The output of the method should be a log file for further use, listing entries with information about custom events of interest occurring in the video, defined by the researcher. When we speak of mobile in this context, we mean devices in terms of tablet or smartphone and visual interactions with the GUI. We do not take other mobile or wearable technology and interaction paradigms into account, this would be beyond the scope of this dissertation. Besides the technical challenges to automatically analyze screen recordings, another major challenge is privacy. Watching over the shoulder while someone is using a smartphone is not only impolite but privacy-invasive. Mobile phones are highly personal and protected devices, making privacy an issue right from the start. The same applies to the permanent recording of the screen, which makes a privacy concept necessary that can convince users to agree on using the method of data collection in a research study, for example. The developed logging mechanisms are applied in the context of education. We are tackling a common challenge when collecting data for learning analytics. Tracking users' or students' traces in digital learning environments that consist of multiple applications in which we cannot simply collect custom log data. The aim is to extend the data collection beyond just LMS log files and to enhance measurements of student activities in learning analytics. The assumption is, that log files from screen recordings can help to draw a bigger and more precise picture of learning processes in digital learning environments and improve the level of detail of student activities we can follow. We need to automatically analyze the screen capture, as the manual analysis is not possible for large amounts of videos of many students. And additionally, privacy is a serious issue in learning analytics as well, which underlines the need for the development of a sustainable approach to the protection of privacy.

The following section introduces more details on how the described goals are divided into individual research questions and gives an overview of how they are addressed. Chapter 3 explains how these questions are answered in the publications of this cumulative dissertation. Figure 1.1 shows by which chapters these questions are approached and how this work is structured.

RQ1: How can we automatically track user behavior in mobile applications based on visual screen output?

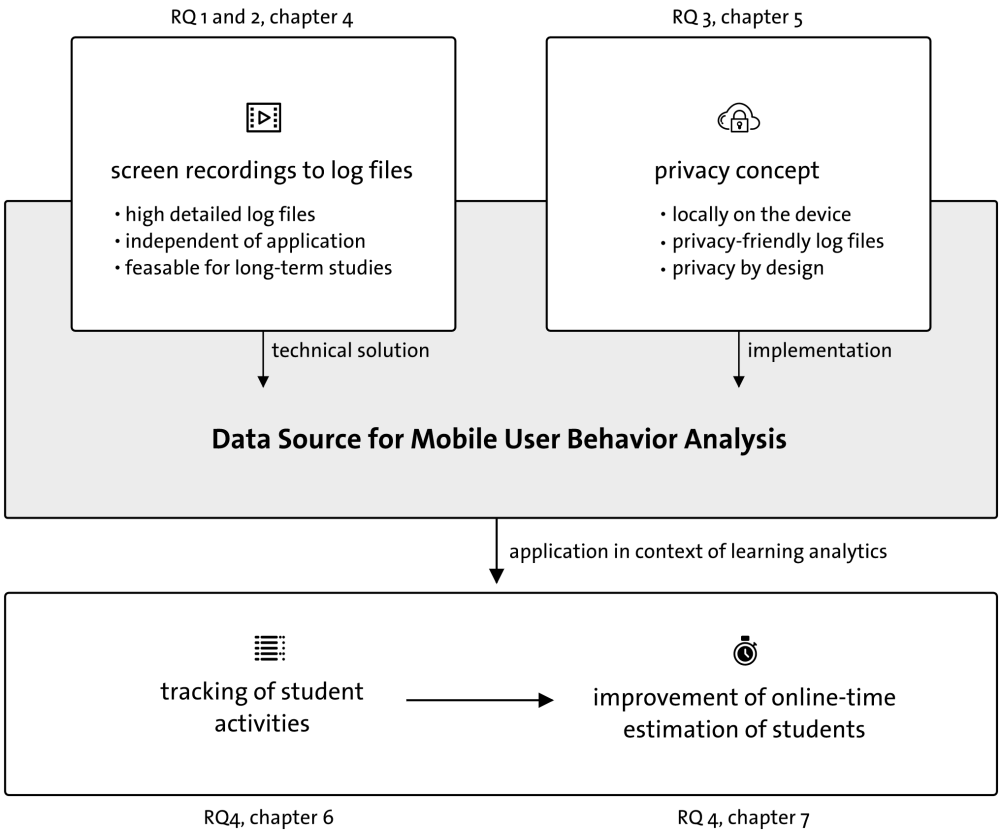

Figure 1.1.: Overview of how the research questions and chapters are organized in this thesis.

This question aims at how we can automatically track events in multiple closed source code mobile applications just based on the visual screen output. This addresses the main technical challenge of the presented approach. It addresses the described main challenges when we want to work with mobile screen recordings: it is a rich empirical data source, independent of applications, but not feasible for long-terms by manual analysis (see section 2.1.3 and section 2.1.4). The focus of this research question is how computer vision and machine learning can be used to generate log files by interpreting the screen output of mobile devices. We show that our chosen approach can generate detailed log data from mobile screen recordings efficiently, by exploiting common characteristics of mobile GUIs. Methodically we evaluate our solution by creating and using a data set of mobile screen recordings to validate the correctness of the resulting log files. We also test different video qualities and how this affects the quality of the resulting log files. For long-term use,

as screen recordings can reach high file sizes, it is important to keep an eye on how much mobile storage the files need.

The following question is closely related and accompanies this first research question.

RQ2: What kind of events can we track and how can these events be defined?

When manually analyzing screen videos, we have a certain goal and are looking for something specific, like an event in the recordings or an action of the user. We need to address how we can transfer this into the technical definition of these events and the process of analyzing screen recordings automatically. This research question aims to explore how events can be defined and what kind of events can be tracked with this approach. By event, we mean the appearance of a certain GUI state on the screen, which we are interested in. For example, an event to track the opening a chat with a person called "Paul', could be defined by the visual appearance of the chat header and fulfillment of the condition of containing the right name. The implementation of computer vision and machine learning depends on the type of events to be tracked and, on the other hand, the type of events that can be tracked is limited by the capabilities of the methods of computer vision and machine learning used. The presented method is independent of the application to be followed, as it only depends on the screen output of the device. The definition of the events of interest must be done in a different way compared to a "classic" approach for log file generation, not by a command in the source code, but as a definition of a GUI state. This raises new questions since the definition of an event based on visual aspects is different from the definition of a log command in an application. We present several different events and demonstrate how we can use them. Furthermore, for logging by commands in the source code of an application, events of interest have to be defined and implemented before starting the data collection. But in some cases, it is not clear at the beginning, which events or log data might be helpful for a certain research goal. With screen recordings, it is possible to redefine events of interest after the screen data has been collected and search again for different events in the video material. This opens the possibility to iteratively improve the event definitions for log entries in order to obtain log data that fits the research goal.

The following question takes privacy in relation to the described logging approach into account.

RQ3: How can we improve the privacy of participants in user studies involving long-term mobile screen recording?

The collected data may contain sensitive information about users. If the user's screen is being recorded, it can be challenging to find participants who are willing to participate in such a research study (see section 2.1.4). This raises the question of how users can stay in control of their data and personal rights, to improve the situation for potential participants. Especially as the intention of the presented work is to encourage further research using the logging method to generate research data. This research question addresses these privacy issues in order to make the approach feasible for future studies with potential participants as well as for researchers. When we speak of improving privacy, we do this not only in a way that is required by laws (e.g.the EU GDPR). But beyond that, a solution that is less privacy-invasive makes it easier for researchers and participants to agree on study conditions and enable research. The search for an ethically accepted solution from both perspectives can enable this type of data collection to be applied in larger contexts. We implement, present and evaluate a privacy concept for our logging approach. The evaluation of the concept is done in a user study with potential participants.

These first three questions form the basis for the application of the approach in the context of education and the fourth research question. This question addresses how the proposed logging approach can improve a common challenge when working with log files in learning analytics: a lack of detail in the data (see section 2.1.3). This research question is being approached in the context of learning analytics but is addressing an issue, which occurs in mobile HCI research and other disciplines as well: unprecise measurement, or lack of data in situations in which we have to guess or estimate what is happening instead of that we can rely on our data. In this case, we address a common measurement in learning analytics: the time students spend using an LMS, which is being estimated from LMS log files, coming with problems and inaccuracies. In short, we know, for example, when a student logged into the LMS, but from that, we can not directly tell, how long the LMS was actually on the screen of the student's device or how long a particular content has been viewed (more details in section 2.2.3).

RQ4: Using the approach in the context of learning analytics: How can we combine details of log files from screen recordings with LMS log files to improve the estimation of the online-time of students in a Learning Management System?

The goal of this question is, to apply and evaluate the developed logging method in learning analytics to collect and analyze data. We tackle a challenge from the data pre-processing steps of LMS log files in learning analytics: estimating how long students are online in an LMS. This is a measurement frequently used in learning analytics for further analysis and in statistical models. However, the definition of how this time is calculated using the LMS log files is not consistent in research and many different ways of calculating it are used, which leads to very different results. The online-time of students is used in models for several predictions in relation to the performance of students. An example is the prediction of probabilities of which students might not successfully complete a course. These students can then be targeted with the aim of improving their situation. By combining common LMS log files with log files generated from screen recordings, a further understanding of how the calculation of online time can be improved is gained. Although this question is very specific, it stands as an example of a more general question in HCI: how can we achieve higher precision and fill in "blind spots" when working with log files which do not provide these details?

In the next chapter, the background and related work of this work are described. In general, the first part of this dissertation is focusing on generating log files of user actions and interactions based on the visually displayed interface of mobile devices and the development and evaluation of a privacy concept.

In the second part, we present an HCI research scenario in the context of education and how we apply our approach successfully in a learning analytics study, highlighting the strengths of the approach presented in this thesis (see also figure 1.1)

2. Background and Related Work

This part gives an overview of the general background of this dissertation, on collecting data to analyze user behavior on mobile devices. Additionally, each chapter of this thesis has its section on the background and related work. The first section of this chapter covers logging from a mobile HCI perspective and work that involves screen recordings. After that, current technical challenges and efforts in research to enhance participants' privacy in HCI research studies involving screen recordings are described. The second section presents an introduction to learning analytics. The section starts with a definition of learning analytics and proceeds with common data sources in this field and related work on current challenges in data pre-processing for learning analytics.

2.1. Tracking User Behavior on Mobile Devices

The approach of this dissertation of generating data is in the field of mobile HCI and especially in the development of methods for tracking and analysis of interactions. To track user interactions on mobile devices, we need to collect data that meets the requirements of the research objective. For this dissertation, two ways of collecting data on the mobile platform are relevant: log files and screen recordings. For long-term data collection, log files are a common and unobtrusive way of tracking application usage, for example. In comparison, screen recordings can provide a very detailed perspective on user interactions but are often used in rather short-termed settings, because the manual analysis is highly time-consuming.

2.1.1. Log files

Log files are a common and effective instrument to run large-scale studies in HCI. Originally, the basic purpose of log files is to record events while running a computer system, like errors, status, and information on performance (Eick et al., 1994). The focus of log file analysis is on the help and support during software development, the detection of errors, software failure diagnosis, and efficiency problems (Oliner et al., 2012). System or software failure log files are usually the starting point for support engineers to back the point in the software or code which might have led to the problem (Yuan et al., 2011). In general, a log file is a text document, containing an entry in every line. The format and content vary, depending on the goal and purpose of the log files: logs from a webshop might be used to maximize sales, logs from a router might save how many devices are connected to the wifi. Another well-known field of use is online marketing and web server access.

Jansen et. al define a log file as "an electronic record of interactions that have occurred between a system and users of that system" (Jansen et al., 2008, p.2). This definition aims more in the direction of interactions between humans and computers, which is relevant for the perspective of how we look at log files for this thesis. On the mobile platforms, there are several approaches to generate log files about the long-term use of mobile application usage at large-scale and in the wild. Also, mobile system logs are in use for security and system failures. Ribeiro et al. (2020) use Android system logs for intrusion detection, for example. Many research studies on mobile application usage are based on log files for data collection as analyzing log files can be highly useful in order to capture user behavior over the long-term or to track specific events of interest. McMillan et al. (2010) describe how they use app stores to run large-scale studies with thousands of users to overcome the problem of having small sample sizes. They distribute a mobile application to create log data on every participant's device, the log data is sent back to the researchers. Similar to this, a large scale study by Böhmer et al. (2011) analyzes the mobile application usage behavior using log files. They log the application use of over 4,100 participants on Android devices to analyze when, for how long, and which applications are being used.
Another common research objective using log files in mobile HCI are usability problems. For mobile evaluation of usablity Balagtas-Fernandez & Hussmann (2009) show how log files collected by their framework during application usage can help to extract usability problems. Lettner & Holzmann (2012a) go a

similar way and present a toolkit to track user interactions and use the data to automatically find possible usability flaws. The toolkit can be added to Android applications during the development stage.
Do et al. (2011) use log files to track places and social context of smartphone use over nine months including 77 participants and infer patterns of phone application usage. Their logging application saves the used applications, locations, and additionally Bluetooth data, as the number of scanned devices nearby can give an idea of how many other devices (people) are around. Most of "early" smartphone studies use custom programmed logging approaches, which have been developed for very specific studies.
In further studies, research frameworks aimed at reuse in research (e.g. Ferreira et al., 2015). Their AWARE framework for mobile sensing logs a wide range of events, like starting and closing an application, battery status, GPS position, phone calls, just to name a few. This is especially relevant from the point of view that this thesis aims at the development of a general approach of data collection on mobile devices as well and multiple purposes of usage of the collected data.

2.1.2. Screen Recordings

Compared to system log files, screen recordings are a different way of collecting data coming with other characteristics. Screen recordings are basically a video of all onscreen activities, which are rendered to the display. McMillan et al. (2015) describe the recording of the screen as data that supports "analysis of micro-level interactions within and between applications on a device", which means we can follow user interactions very closely in any application. A popular example of the practical use of screen recordings are online tutorials on platforms like YouTube that explain how to use a certain software. These screen recordings are easy to follow and thus can help users to learn how to use a certain application, similar to a step by step instruction (see e.g. Lafreniere et al., 2014). Before the spread of smartphones and tablets, screen recordings had already played a role in desktop computing. The number of publications per year involving screen recordings has increased considerably since the beginning of the 2000s, from below ten to over 70 per year in 2019 (searching the Scopus database for TITLE-ABS-KEY("screen recording"OR"screen capture"OR"screen video")). Screen recordings are related to video-based observation and documentation. Before the technical improvements enabled com-

putational light-weight screen recording in the background, video cameras have been used to conduct interaction analysis of technology (e.g. Ruhleder & Jordan, 1997). Ruhleder & Jordan (1997) use video to capture complex activities in workspaces, to extent traditional ethnographic methods. Similar goals have been pursued later, using screen recordings. Tang et al. (2006) use long-term screen recording to unobtrusively collect data on how teams work together on multiple computers and coordinate work. They state that this data source makes it easy to follow users' interactions and to collect rich empirical data in the field. Imler & Eichelberger (2011) follow a comparable approach and capture the screen videos of how students use the library's databases at the computers of the library and evaluate and discuss screen recordings as a way to capture human-computer interaction. They use coding software to analyze the material manually and value the source of data as helpful and inexpensive.

Another frequent use for screen recordings, which emerged early and is related to the described observational interaction analysis, are usability studies in laboratory settings (e.g. Fast, 2002). A common concept is to record the screen together with the surrounding audio and a think-aloud protocol (Kushniruk & Borycki, 2006). Kushniruk & Patel (2004) present a setup of using screen recordings, think aloud and video observations of the user to evaluate the usability of health software applications. Similar setups are currently still in use in recent usability studies (see e.g. Planas & Cabot, 2020). With fewer limitations due to improved hardware, recording the screen is not resource-intensive anymore. Industry usage of mobile screen recordings is usually related to remote usability testing. Companies offer developers to have their applications tested and receive screen recordings together with parallel audio and video recordings of the user(tester) through the camera of the phone to get feedback during the development without the effort of having to organize a user study (see e.g. lookback.io, 2020; userfeel.com, 2020).

With the emergence of smartphones, mobile screen recordings moved into focus for the application of screen recordings in usability studies. Liang et al. (2011) show how mobile screen recordings can be used for remote usability studies. Even for mobile devices, recording the screen is not resource-intensive anymore for modern devices (Krieter & Breiter, 2018a). In comparison to filming the device with an external camera, an obvious advantage is, that a screen recording can not be occluded by the user's hand, which is especially important in the mobile context as most smartphones or tablet devices

have touch screens. Borycki et al. (2015) state that mobile screen recordings are the least obtrusive way to evaluate mobile usability.

As the main contribution of this research work is to automate the analysis of mobile screen recordings to expand the use of this rich data source to a long time span we focus on projects which involve mobile long-term screen recordings, which distinguishes them from the use of screen recordings for short-term usability studies. There are rare research studies that use manual analysis of screen videos in order to investigate mobile usage behavior and show how to utilize screen recordings as a detailed data source to learn about how users interact with their devices. Brown et. al collected screen recordings, audio, GPS location, and application launches of iPhone users and let the participants annotate the video material in form of small diary entries for later analysis by the researchers (Brown et al., 2014). Related to this approach is the so-called *in vivo* method combines logs of system data like GPS position, active application, and additionally video recording of the screen and the phone's camera and audio of the surrounding noise and voices (McMillan et al., 2015). Reeves et al. (2019) present a framework for collecting and analyzing sequences of screenshots they call 'screenome'. They collect screenshots on mobile and desktop devices at five-second intervals. They are trying to capture screenshots of all screen time on multiple devices together, of one person. Although this is different to screen recording which covers all interactions happening on the screen, instead of only one still image every five seconds, they use this way of collecting data for long-terms. This relates the work to this thesis, even if single screenshots are the basis. They transfer these screenshots to a server and extract text and image information using OCR and computer vision to organize the results in a searchable database. In terms of automating the processing of screen recordings, there are previous studies on the desktop platform that guide the development of the logging approach presented in this thesis. Frisson et al. (2016) propose an automatic screencast annotating system using computer vision techniques and OCR. Comfortable from a researcher perspective, they present GUI editor to define events based on screenshots, comparable to an annotation tool (Frisson et al., 2016).

The scope of application for this thesis is the use of screen recordings to track and analyze mobile human-computer interaction. In addition to this more general overview on screen recording, the related working sections of the publications in the first part of this dissertation give a specific overview of earlier work on each individual contribution (see section 4.2 and 5.2). We build on existing methods and address previous efforts in mobile HCI research,

which aim to further expand the use of screenshots or screen captures, and several approaches strengthen screen output as a highly valuable data source.

2.1.3. Technical Challenges

Both types of data collection and analysis, whether using log files or screen recordings, are associated with different technical challenges.

A challenge when working with log files is that most log entries lack some of the context in which they occur. A web server log entry tells when a person has visited a certain website. However, it does not say how long the person looked at the page, how the scrolling happened, or whether the person immediately switched to another program. Addressing the missing context when debugging software, Yuan et al. (2011) present the "Log enhancer", an approach to add more information to log messages in software to enhance and reduce the overhead when diagnosing software problems. This is an example for the mentioned aspect, that log files might contain a smaller picture of what is happening, making a detailed analysis a challenge. In this case, the tool adds information, that programmers might have added to the log message to easier solve problems. Another phenomenon is that log files consist of records and information that are not relevant to the purpose for which they are used and that filtering and pre-processing of the data is necessary. Eick et al. (1994) speak of "log noise", which are unimportant messages in the log files which do not help on finding a certain problem. On the other hand, non-obvious problems might also be hidden in this noise, which can make the analysis of log data in general challenging.

The missing context is especially important when using log files in mobile HCI research. This results in some cases from using system log files for something other than what they are intended for: user research instead of monitoring and debugging software. When using log files to research mobile user behavior, several challenges frequently occur. Most mobile logs are limited to rather general system events like which application is present or if the screen is turned on or off, for example. Usually, to generate log files about what is happening inside a certain application, we have to implement log commands in the source code of this application or the operating system. McMillan et al. (2015) state that automated logging removes a lot of the context information in which the usage of the mobile device occurs. Thus, mobile-log approaches are able to answer a variety of questions about the use of smartphones and

applications and to create data sets for current large-scale research. However, they are limited in the way that they cannot take into account what users actually do within the applications (Böhmer et al., 2011). Regarding this limitation Böhmer et al. (2011) mention in their large-scale study on mobile application usage, based on log files, that the details on what is happening inside of an application are not available from these logs. As an example, they name a web browser, an application that can be used for multiple purposes, like looking something up in a dictionary to checking public transport. This limits the explanatory power of these log files. Frameworks like AWARE, solve the lack of context to some extend, by making the tracking of many mobile sensors and data points possible within one consistent data format. Through this combination of many phone sensors, the user context can be inferred to some extend. An example could be to estimate how the user holds the phone (gyroscope) or whether the user is in a car (GPS) (Ferreira et al., 2015).

There are different tools, frameworks, and custom solutions to collect log data on mobile devices. For example, many applications offer APIs to collect data (e.g. WhatsApp, Instagram, SMS service), but it is quite costly to bring all these different data sources together, as most of them come in different formats and are not customizable in what they exactly track or provide. It is not easy to combine and analyze many different sources and formats of data together. Reeves et al. (2019) describe this as a problem for their "screenome" project and as a reason why they choose to take screenshots every five seconds and use OCR on the images instead of collecting their data via APIs of several different applications, to collect data on social digital activities of their participants.

From a technical perspective, screen recordings as a data source for mobile tracking are much more resource-intensive than storing system events in a log file. At the same time, resources on mobile devices are usually more limited compared to server solutions or desktop computers. Permanent video recording needs storage capacity, although this limitation is becoming less important due to increasing local storage (McMillan et al., 2015). From the dataset we use for the first publication of this dissertation we know, that recording the phone's screen in high quality for 24 hours which results in about one hour of screen on time, we get video files of 2,2 GB (see table 4.4.1 for an overview of different video settings and filesize). In terms of modern mobile device storage, even high-quality video is unproblematic, but transferring large video files to research servers is still a challenge.

An obvious challenge when working with screen recordings is that the manual analysis of videos takes a considerable amount of time. Depending on the task or research question, the time required for manual review and analysis of screen videos can vary widely. The previously presented work which involves long term screen recordings has different approaches to address this. Brown et al. (2014) let their participants label the recordings in the form of a video diary, which makes the categorization, interpretation, and finding of recordings easier. McMillan et al. (2015) triangulate the data with other sources such as log files to cross-check the screen recordings with additional information. One example is location data. If only recordings at a specific location are of interest, this is a way to reduce the number of videos to be analyzed and to speed up the process.

When it comes to analyzing screen recordings automatically, the computational cost is high. Frisson et al. (2016) state that using computer vision for processing large amounts of video material limits the possibilities of their approach and mention speeding up the processing of screen recordings as a challenge to address. Additionally, they see possible errors in the detection of computer vision as a problem. The tool "Sikuli Test" (Chang et al., 2010) is an approach to test desktop GUIs automatically using computer vision methods, which does not produce log files, but faces and solves a similar problem: recognizing certain events in the user interface and parts in the GUI to act upon it. They make a lot of use of the computer vision method template matching to find image parts in another image (video frame), which takes significant time but works reliably. Their "Sikuli" framework is still relevant for similar current projects on GUI automation (see e.g. Ramler et al., 2020).

Related challenges are reported by Reeves et al. (2019), who in part work with the labeling of screenshots to train machine learning models for recognizing certain applications. This comes with two challenges. First, to train their machine learning model, they need labeled training data. They manually label screenshots to create a data set to use as ground truth for their machine learning pipeline, which takes a lot of human labeling work. Second, the training of models needs time and computational power. The results of their machine learning model on labeling screenshots are according to them greater than 85% in correctness, which still is a challenge as a considerable amount of labels are wrong.

Privacy can also be considered from a technical perspective, but these issues will be discussed separately in the next section, as they are particularly relevant to a large part of this work.

2.1.4. Privacy Challenges

While it is technically possible to collect more data than ever before, it is still a challenge to enforce user's rights of privacy and control over their data and at the same time enable researchers to collect necessary data. It is not easy to transfer, social, ethical, and legal aspects into system requirements.

There is a discrepancy between privacy interests and personal rights of users and the obligation to keep research data and make it accessible for further research purposes or reproduction of results. Gebel et al. (2015) describe this aspect of research data management using the example of qualitative interviews. Complete anonymization to protect the personal rights of participants generating this data is difficult and changes the data significantly, which reduces the usefulness and reproducibility of results for others. Often data is reused for new research purposes other than the one initial purpose, leading to important contributions, as for example Lowrance (2003) states. Primary medical data is often stored in electronic databases and reused, to conduct further research, for example on epidemics, patterns of occurrence, or evaluation of healthcare. There are efforts to develop methods making data accessible while still keeping it confidential to prevent inferring of individuals from research data. Boker et al. (2015) present an approach based on data that is only saved privately on the user's device. All calculations for research involving the participants' data is done locally on their devices, the research side only receives the result. They improve privacy through a decentralized structure of participant data. A similar step is also an important part of the privacy concept presented in chapter five.

This raises the question, how participants in research studies can be aware of which data they are sharing for what purpose and how they can control what they share. On the other hand, it is important to enable research and to maintain a balance between the effort and benefits of privacy ambitions, for participants and researchers alike. This is a frequent challenge in current HCI research on ethics and privacy (e.g. Brown et al., 2016). Informed consent is necessary for most HCI studies unless they are completely anonymous. Hence, having a concept of privacy that is able to convince participants, and not just fulfilling the legal expectations, should be the goal of ethical research. Brown et al. (2016) discuss these issues in a position paper on HCI research ethics and state that "not harming or burdening participants is more important than teaching them the intricacies of academic studies", advocating a balance between privacy in practice and research interest.

The concept of Privacy by Design emerged in the '90s and contains seven principles to take into account: *Proactive not reactive, Privacy as the default setting, Privacy embedded into the design, Full functionality, End-to-end security, Visibility and transparency, Respect for user privacy* (Langheinrich, 2001; Cavoukian, 2011). The idea is to make Privacy by Design a standard to improve privacy by considering it while developing a product and even before. In a position paper, Schaar (2010) emphasizes that it is less time-consuming to address data protection at an early stage during development or in the planning phase and not only afterward as a "patch" to make the software compliant with data privacy regulations. Along with the guidelines of Privacy by Design, Gürses et al. (2011) present two case studies from software engineering, an ePetition system and a road toll system, and state that there is no "one-way" solution to privacy, which makes the implementation of privacy very complex. Rubenstein and Good notice that although companies and regulators agree on Privacy by Design as an important aspect to incorporate, but struggle on how this can be achieved in detail (Rubenstein & Good, 2013, p.1349). More indications are pointing in this direction. Ayalon et al. conducted a study on how developers make decisions about privacy because approaches like Privacy by Design are not a common principle in companies and development departments (Ayalon et al., 2017). Related to Privacy by Design is the "privacy-by-architecture" approach that is defined by trying to implement the minimization of the collection of personal data that can be identified and focuses on its anonymization (Spiekermann & Cranor, 2009). In addition, the processing and storing of personal data is strongly preferred on the client-side instead of network storage, as Spiekermann & Cranor (2009) emphasize. They advise this approach over the "privacy-by-policy" approach that emphasizes the notice and choice for users on which of their personal data is used but does not implement privacy by design or default.

While anonymization can work for log files, the case is different when it comes to permanently recording the screens of participants. The anonymization of log files is different compared to screen recordings. Web server log files, for example, usually save which page has been viewed, the IP-address, and time and date. This strict format makes it possible to remove the IP address, which is an efficient step in the direction of anonymizing these kinds of files. Screen recordings have obvious privacy and ethical problems when used for data collection in user studies. As they reveal everything a user sees while interacting with a device, many abuse scenarios are possible. From an ethical point of view, it is therefore of interest to consider whether such data

collection is justified by the benefits of the research. From a privacy perspective, the automatic analysis has the advantage that we only search for specific events. In comparison, the manual analysis will always lead to the problem that a person watching the recording will see more than just the actions of interest. However, this alone is not a sufficient data privacy concept to use screen recordings for long-term user studies in non-laboratory settings. The fact that screen recordings are privacy-invasive is a reason for participants or potential participants to leave or not join a user study which involves the long-term recording of the screen. This is an important challenge when working with screen recordings in user studies. Tang et al. (2006) use long-term screen recordings and conduct interviews with their participants on how they perceived this privacy-invasive way of data collection. They conclude, that some participants declined to join the study due to privacy considerations. A similar experience is reported by Reeves et al. (2019), who collect screenshots of their participants' devices every five seconds. A study, which also works with screenshots at a five-second interval to collect data on desktop computers over two weeks, mentions, that one of 14 participants left the study due to privacy concerns Vuong et al. (2017).

We address these privacy issues related to screen recording in the first part of this dissertation (see section 5.2) and present a solution approaching this problem. Most previous work using screen recordings is aware of these privacy issues but still lacks to address these to enhance the situation for both: participants and researchers.

2.2. Tracking for Learning Analytics

Tracking for learning analytics and HCI research has many similarities. Both disciplines share methods and do research on user behavior and modeling. They use similar data sources to do so, but with different research objectives. As mentioned in the introduction, this field is a promising domain for the application of the logging approach presented in this thesis, as we can directly address challenges in data pre-processing for learning analytics. In mobile learning analytics, logging mobile device user behavior is done to analyze or predict the learning outcomes of students. learning analytics is often used to predict learners' performance or problems and to intervene. This results in privacy being an important issue in learning analytics as well, as these pre-

dictions are made on a personal and individual basis for students and teachers. This part starts with a definition of learning analytics and common data sources, followed by an overview of current challenges in processing data for learning analytics.

2.2.1. Definition of Learning Analytics

With the use of digital devices and services for learning, more and more data related to education is available. Analyzing this data enables new ways, of how we can look at learning processes, which happen in digital learning environments. The first international conference in 2011 on learning analytics and Knowlege defines learning analytics as the measurement, collection, analysis, and reporting of data about learners and their contexts, for purposes of understanding and optimizing learning and environments in which it occurs"(LAK-Conference, 2011). Siemens (2013) considers this definition as the most common and adopted definition and additionally sees Cooper's definition of analytics drawing from business intelligence as helpful: "Analytics is the process of developing actionable insights through problem definition and the application of statistical models and analysis against existing and/or simulated future data." (Cooper et al., 2012). Papamitsiou & Economides (2014) follow this and state that learning analytics are related to ´´business intelligence, web analytics, academic analytics, action analytics, and predictive analytics". Ifenthaler & Schumacher (2016) summarize learning analytics make use of data regarding individual student characteristics, like prior knowledge and academic achievements. Additionally, activities in the learning environments, like page views and upload or download activities, for example. Also benchmarks from the course of studies, like learning outcomes, and the communication and interaction in their network of other students and teachers.

Learning analytics consists not only of the technical realization of collection data and detect patterns but to generate insights for multiple stakeholders, like students, teachers, and university staff, as Drachsler & Greller (2012) state. Since learning analytics is about learning, the involvement of these stakeholders is particularly important and the discipline must show that it can have a long-term practical impact on students and on teaching Gašević et al. (2015). A common application is the approach of identifying or predicting students who have problems in coping with a course or their studies at an early stage

and offering support through appropriate interventions (see Papamitsiou & Economides, 2014).

The field of learning analytics is closely related to Educational Data Mining, which focuses more on the development and application of methods to analyze large educational datasets in order to recognize patterns (Romero & Ventura, 2013; Papamitsiou & Economides, 2014). Both fields have overlaps and "constitute an ecosystem of methods and techniques" (Papamitsiou & Economides, 2014) to encourage thinking about learning processes and improve educational environments. As a result, the goals of both disciplines are similar. In contrast to educational data mining, learning analytics focuses more on sensemaking and deriving action from data, while educational data mining is more concerned with methods for data analysis of educational data (Siemens, 2013).

2.2.2. Data Sources for Learning Analytics

Like in HCI research, learning analytics studies are often working with log files as well and facing similar problems. The most commonly used data source for learning analytics are log files from LMS (Papamitsiou & Economides, 2014; Pardo & Kloos, 2011). LMS have become an integral part of the university context, schools, and business. They form the information technology basis for the organization of courses, communication between students and lecturers, and between each other, right up to the handling of online tutorials and examinations. Due to the high prevalence of LMS, most log file analyses in learning analytics are performed with data from these systems Pardo & Kloos (2011). When used by students and lecturers, data traces are generated which, stored in the form of log files, are used for the analysis and evaluation of a wide range of questions. A common application is to identify or predict students who have problems in mastering a course or their study program and to offer support by appropriate actions (see Papamitsiou & Economides, 2014). Comparable to challenges described in the previous sections on tracking in mobile HCI research, data sources for log files used in learning analytics struggle to provide a whole picture of what is happening in a digital learning environment. Learning analytics have been described as "LMS-centric"(see Pardo & Kloos, 2011). Log files from an LMS are often just a small piece of all interactions which happen in a digital learning environment. Often, communication between peers happens through existing mobile chat or social media applica-

tions and not through the communication features of the used LMS (Pardo & Kloos, 2011; Dyment et al., 2020). This results in many studies that use additional data sources. The second part of this thesis presents the approach of this thesis based screen recordings as a data source from the learning analytics perspective and shows how this data source can be used to collect data beyond LMS log files. Section 6.2.1 and 7.2.1 give an show related work on data sources in learning analytics outside the LMS. This paragraph presents an overview. There are examples of using data sources outside an LMS for LA in the context of learning programming. Pardo & Kloos (2011) presents an approach using virtual machines to log a number of events outside of an LMS. They provide pre-configured virtual machines to students of a programming class that the students run on their personal computers. The machines monitor a couple of events like power-up and shut down, start and use of selected tools and commands, and browser history. They show that a significant amount of relevant interactions happen outside of the sphere of the used LMS. Related to that, Blikstein is using a dataset of a three-week student assignment on programming using a programming environment that logs many users' interactions, such as keystrokes, clicks, variables changes, and changes in the source code (Blikstein, 2011). He shows, how this data can be used to find certain events in the process of programming and suggests identifying situations in which students might need help. Fernandes-Medina et al. use compile messages as a data source and analyze the work of students to report on the individual and comparative progress of learning (Fernandez-Medina et al., 2013). They use the results to inform students about their learning process. Another recent related approach to this is pursued by Öztürk et al. by developing a web-based programming environment for novice students to collect data Öztürk et al. (2018). By identifying metrics for student performance they use this data to predict students at risk to drop out at an early course stage. Another attempt to gather data beyond the LMS is introduced by Kitto et al. (2015) utilizing data from social media combined with LMS data. They present an open source toolkit accessing six social media APIs and save it in a standardized notation for further analysis. Their focus is to gather data from multiple sources in a uniform way at large-scale, but also concerns regarding related privacy and data ownership issues. Another study by Tempelaar et al. (2015) explores and compares the predictive power of different learning analytics data sources. Therefore, they collect self-reported data, LMS data, and e-tutorial data of formative assessments. In the conducted study, data from 922 mathematics and statistics students is being collected. They mention that the use of only LMS

data does not have substantial predictive power in their study. They find that the use of formative computer-assisted assessments is a good predictor for detecting underperforming students.

It is not decided within the learning analytics community which student interactions in a digital learning environment are crucial for effective learning (Agudo-Peregrina et al., 2014). This makes it important to ask which data points are worth tracking and which can be technically tracked and used for good predictions and analytics. It is a current challenge to re-evaluate measures of online activities of students, because in some cases it is unclear, what is actually measured. The relevant challenges for this thesis are addressed in the next section.

2.2.3. Challenges in Data Pre-processing

Typically, various data points are included in order to be able to make predictions about student performance factors using statistical models. Frequently used data points are, for example, the length of time students spend in the system, or the number of clicks, and logins in a given period. You (2016) presented a study on which indicators from LMS usage are significant to predict the success of students in an online course. The study included 530 students and showed that regular participation, the late submission of assignments, and the number of sessions are decisive. A session means, in this case, the frequency of logins to the LMS. A study by Kovanović et al. (2015) takes a closer look at another measure which is derived from LMS data. They explore the time-on-task estimation of students using different methods to calculate this time and its effect on commonly adopted models. Their results show, that the chosen time-on-task estimation methods have a significant effect on the outcome of statistical models. They intend to bring more awareness of this and similar processes in learning analytics, as the different methods for estimation lead to different assumptions on student performance. Conijn et al. (2017) compare the results predicting student performance of 17 blended learning courses of the LMS moodle of one institution. The courses consist of 4,989 students. They use common predictive LMS variables from the literature (e.g. clicks, session, online-time, etc.) and conclude that the portability of predictive models of the courses is low and varies strongly, which questions the validity. Statistical models that worked well for one course did work for

other courses. They emphasize the need to include additional data sources beyond just LMS log files.

These studies raise a central question when preparing data for use in statistical models: What do we actually measure? In many cases, certain measurements are derived from the log files of LMS, which always involves a certain degree of uncertainty. This can be seen, for example, in the calculation of the time, a student spends in an LMS. In the literature, there are numerous definitions used to calculate this time. Marquardt et al. (2004) specify a session quite vague as a set of user's accesses during a visit. Another definition by Ba-Omar et al. (2007) describes a session to calculate the online-time as a learner's access to the system, ending with a timeout after 30 minutes to start a new session. del Valle & Duffy (2009) also use a similar definition and a 30-minute timeout for the last activity in the system. A study by Munk & Drlík (2011) identifies a session "as delimited series of clicks realized in the defined time" and use a timeout of 15 minutes to start a new session. Sael et al. (2013) calculate based on all user interactions between login and logout, which leaves open how they define the end of a session if the user does not actively log out. Zacharis (2015) use a 40-minute timeout for a session, which contains all user action between login and logout or a timeout of the session. The already mentioned study by Conijn et al. (2017) uses a similar definition, also with a 40-minute timeout and use the time between first and last click to calculate the online-time of the students.

These definitions differ mainly in the length of the time window, which may lie between two clicks, in order to calculate the end and length of stay in the system based on a series of clicks (see also table 7.1). This leads to very different calculations, depending on the definition used. Kovanović et al. (2015) show that these differences in calculation can, in turn, have a strong impact on the predictions of statistical models based on this data and thus, for example, on the identification of students who have problems in completing their studies. Apart from rather simple derivations such as duration of stay or number of clicks, more complex theoretical constructs such as student motivation cannot be derived from pure LMS log files, as Conijn et al. (2017) state. Dyment et al. (2020) present findings from an interview study pointing in the direction that the engagement of students in online courses is not measurable with just LMS data, because many interactions and relevant activities are happening outside of the closed learning environment. The problems faced by the learning analytics community appear in a similar way in HCI research. A study by (Ernala et al., 2020) is using a dataset from Facebook with detailed server logs

from the social network that is used to estimate the time users spent on Facebook. They state that even with these detailed logs from their own servers it remains complex and sometimes inaccurate to estimate the time users exactly spent on the platform. This underlines the complexity of online time estimation in general. Section 7.2.2 gives more details data pre-processing for student online time estimation in learning analytics.

In the second part of this dissertation, we address these issues in the data pre-processing and show how the approach developed and described in this work can be used to generate data for LA, with a focus on the precise online-time estimation of students (see section 7.2.2).

2.3. Summary

From the described findings and challenges in previous research on collecting data on user behavior in the fields of HCI and learning analytics, we can summarize several aspects which are relevant for the work presented in this dissertation. Some conclusions we can draw from the previous work are listed below.

- Mobile HCI research using log files (e.g. based on system events) struggles on taking into account what users are doing inside of applications because they lack these details but are highly suitable for long-term studies. Similar issues are reported by previous work in learning analytics working with log files, mostly from LMS. These LMS log files lack details for important measurements in this domain, for example, the online time of students. Due to these limitations, previous work in learning analytics advocates the use of additional data sources to extend the data basis for analytics from just LMS log files.

- Screen recordings provide details of all activities that occur on the screen at a micro-level but are not feasible for long-term studies because they usually have to be analyzed manually, which is time-consuming. Multiple studies emphasize this data source and its level of detail and independency of applications. Screen recordings are used often for usability tests, interactional analysis in the tradition of video-based observations, or as an additional data source in combination with log files, for example.

- The strong argument that screen recordings reveal everything on the user's screen results in a privacy problem, which is beside the time-consuming analysis, another major limitation. Previous research working with screen recordings reports this as a challenge when recruiting and working with participants in user studies because participants are skeptical as recording the screen is highly privacy-invasive.

From this, we can conclude, that a way to collect data on user behavior at high detail, regardless of which application, long-term, and privacy-friendly on mobile devices is highly beneficial for further research in the fields of mobile HCI and learning analytics. This forms the basis for the research presented in the next chapters.

3. Publications and Thesis Structure

This research work consists of five peer-reviewed original publications in international conferences, a journal, and an article in an edited collection. This section explains how these are integrated into this dissertation and which research questions the individual publications address and how. In general, this thesis contains two main parts. The first part deals with the technical realization and evaluation of the proposed logging method and privacy-related aspects in terms of applying this approach in user studies. The second part addresses the fourth research question and how the developed logging approach can be used to generate data in a research study in the context of learning analytics and how we can extend LMS log files with log data generated from mobile screen recordings.

Part I: Technical Solution, Privacy and Evaluation

This section addresses the first two research questions and consists of two publications. The following article addresses the first and second research questions and describes how mobile screen recordings can be analyzed automatically to track mobile application usage and how events can be defined. This paper has been **nominated for the best paper award** and received the **Honorable Mention award** at the MobileHCI 2018. I developed the idea for this paper and was responsible for the technical implementation and the conduct of the study and experiments as well as for the analysis and interpretation of the results. The paper was written by me, with feedback and support from Andreas Breiter.

> **Philipp Krieter** and Andreas Breiter (2018). Analyzing mobile application usage: generating log files from mobile screen recordings. In Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services (**MobileHCI '18**). ACM, New York, NY, USA
> DOI: https://doi.org/10.1145/3229434.3229450

Abstract: Logging mobile application usage on smartphones is limited to rather general system events unless one has access to the operating

system's or applications' source code. In this paper, we present a method for analyzing mobile application usage in detail by generating log files based on mobile screen output. We are combining long-term log file analysis and short-term screen recording analysis by utilizing existing computer vision and machine learning methods. To validate the log results of our approach and implementation we collect 118 sample screen recordings of phone usage sessions and evaluate the resulting log file manually. Besides that, we explore the performance of our approach with different video quality parameters: frame rate and bit rate. We show that our method provides detailed data about application use and can work with low-quality video under certain circumstances.

The next publication does not only put privacy in focus but the technical scalability of this logging approach for long-term HCI research studies. This publication is addressing the question of how we can improve the privacy of participants in user studies involving long-term mobile screen recording.

> **Philipp Krieter**. (2019).Can I record your screen? Mobile screen recordings as a long-term data source for user studies. In Proceedings of the 18th International Conference on Mobile and Ubiquitous Multimedia (**MUM '19**). ACM, New York, NY.
> DOI: https://doi.org/10.1145/3365610.3365618

Abstract: Mobile screen recordings are a rich data source to follow user interactions, but at the same time are highly privacy-invasive and manual analysis is time-consuming. This data source is mostly used in small, short-term user studies. Cloud-based approaches can automate the screen recording analysis but lack addressing that recordings contain sensitive private data. This makes ist hard to find participants for user studies involving long-term screen recording. In this paper, we present an approach to automatically analyzing mobile screen recordings on the user's mobile device that respects privacy and makes the approach scalable. We evaluate our privacy concept in a survey study of 35 participants, which indicates our concept increases the willingness of participants to participate in research studies using this method. For technical validation, we carry out test runs on several different devices and show that our approach can provide long-term results reliably in a pilot study.

Part II: Application of the Approach in the Context of Education

The second part of this dissertation addresses how we can apply the presented approach in the context of education, for learning analytics. This part consists of two publications. The following paper considers the logging approach as a data source from the perspective of how we can track students in digital learning environments regardless of which application or an LMS.

> **Philipp Krieter** and Andreas Breiter. 2018. Track every move of your students: log files for Learning Analytics from mobile screen recordings. In the 16th E-Learning Conference in Computer Science (**DeLFI 2018**). Bonn: German Informatics Society e.V.. p. 231-242. https://dl.gi.de/handle/20.500.12116/21042

With the feedback from Andreas Breiter I identified the need in learning analytics for additional data sources to capture student activities outside the LMS and developed the idea and concept for this paper. The paper was written by me with comments and advice from Andreas Breiter.
Abstract: One of the main data sources for learning analytics are Learning Management Systems (LMS). These log files are limited though to interactions within the LMS and cannot take into account interactions of students in other applications and software in a digital learning environment. In this paper, we present an approach for generating log files based on mobile screen recordings as a data source for learning analytics. Logging mobile application usage is limited to rather general system events unless you have access to the source code of the operating system or applications. To address this we generate log files from mobile screen recordings by applying computer vision and machine learning methods to detect individually defined events. In closing, we discuss how these log files can be used as a data source for learning analytics and relevant ethical concerns.

The fourth paper is focusing on a practical example and use of research data generated using our logging approach in a learning analytics study and emphasizes and demonstrates the strength of this approach. In learning analytics, missing details of log files and the validation of measurements are reported challenges. Our way of generating log data about learning from screen recordings addresses both of these issues. We aim at providing detailed data of all activities in a digital learning environment, independent of application or LMS. Additionally, we show how we can use this additional data to evaluate existing measurements in learning analytics. We present results that show that common

online-time time estimations do not work for the actual online-time of our participants in a popular LMS.

> **Philipp Krieter**. 2020. Are You still there? Estimating Students' LMS Online-time by combining Log Files and Screen Recordings. **IEEE Transactions on Learning Technologies**. Under review (submitted 6th of January 2020).

Abstract: The time students spend in Learning Management systems (LMS) is an important measurement in learning analytics (LA). One of the most common data sources are log files from LMSs, which do not directly reveal the online time, the duration of which needs to be estimated. As this measurement has great impact on the results of statistical models in LA, its estimation is crucial. In the literature, there are many strategies for estimating the duration, which do not represent the actual online time of the students. We combine LMS log files of our students with parallel screen recordings and automatically analyze for how long the LMS is present in the video. We visualize the results and show that common online time estimation strategies do not represent the online time for our students accurately. By using modified online time estimation methods, we find estimations that fit the data of our students better on an individual basis.

Although the fifth publication is thematically anchored in the second part, it forms the basis for the background and related work of learning analytics at the beginning of the thesis. This article has been published in an edited collection. The background section (2.2) of this thesis, which gives an overview of the definition, data sources, and pre-processing steps in learning analytics, consists of parts of this publication. Andreas Breiter supervised the project and provided feedback in all phases of the creation of this publication. I developed the idea for the data preparation, analysis, visualization and interpretation of the results. The original publication has been published in German language, the title translates to "Digital traces of students in virtual learning environments".

> **Philipp Krieter,** Andreas Breiter. 2020. Digitale Spuren von Studierenden in virtuellen Lernumgebungen. In Hofhues, S., Schiefner-Rohs, M., Aßmann, S. & Brahm, T.. Studierende - Medien - Universität. Einblicke in studentische Medienwelten. Münster: Waxmann. p. 131-152. ISBN: 978-3-8309-4049-4

Abstract (translation): This paper describes the results and the process of log file analysis in the Your(r)Study project, which aims to combine the qualitative data of the project with the quantitative log data of a Learning Management System. By means of descriptive methods and clustering the log data will be examined with regard to the questions of the project. In addition, there is a critical discussion of the limits of a log file analysis in relation to the goals.

Part I.

Technical Solution, Privacy and Evaluation

4. From Mobile Screen Recodings to Log Files

This chapter has been published with the title "Analyzing Mobile Application Usage: Generating Log Files from Mobile Screen Recordings" in the Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services 2018 (MobileHCI'18) published by the ACM.

4.1. Introduction

Several insights from research into the use of mobile applications are based on log files from data collection applications or frameworks, as analyzing log files can be very useful in order to capture user behavior over the long term or to track specific events or usability problems. (Böhmer et al., 2011; van Berkel et al., 2016; Do et al., 2011; Balagtas-Fernandez & Hussmann, 2009; Lettner & Holzmann, 2012b,a). Most logs are limited to rather general system events (McMillan et al., 2015) like which application is present or if the screen is turned on or off. These approaches are able to answer a variety of questions about the use of smartphones and applications, but cannot take into account what users actually do within applications (Böhmer et al., 2011). A user that opens Whatsapp could just chat with another person, but could also be in an audio or video call, update his or her status, take a picture, video or voice message and send it to a group chat. This is making the analysis of in-detail mobile application usage difficult, as most logs can not provide deep in-app insights of events or interactions inside applications without modifying the apps' source code.

In contrast to log files, the analysis of mobile screen recordings provides a different view of user interaction. The strong focus on GUIs in HCI leads to the fact, that the interpretation of screen recordings enables understanding almost any action, behavior or task the user is pursuing while using a computer system as a GUI always represents the current state of a system, based

on user and system events (Aho, Kanstren, et al., 2014). One widely used possibility resulting from this is the recording and analysis of screen videos to draw conclusions about how users interact with software or to evaluate usability or user experience. The analysis of screencasts can provide deep qualitative and quantitative insights into how humans use and interact with computer systems (Kushniruk & Patel, 2004; Haak et al., 2003; Barnum, 2010; Lazar et al., 2017). Hence it is a time-consuming method: the analysis of screen recordings to find certain events or actions of interest is not suitable for long-term interpretation of hours, days or even years of screen recordings. This limits the analysis of screen recordings to relatively short time spans. Currently most user experience evaluation methods are focused on short-term evaluations (Vermeeren et al., 2010).

Knowledge of how users interact with applications is essential for the development of software products. Therefore we combine two common existing approaches to generate data about user interaction: log files and screen recordings. The goal is to utilize the strengths of both methods to gain meaningful data for further analysis.

In this paper, we propose a method for analyzing mobile application usage in high detail by generating log files based on mobile screen recordings through applying computer vision and machine learning methods. This approach has some major advantages. The method is not dependent on implementing log commands in applications' source code and is for that reason applicable to "all" application and systems events that occur on the user's mobile screen, extending the logging capabilities from rather general system information to in-app interactions. It is possible to track multiple mobile applications using the same method. The definition of log events of interest can be done even after collecting data, as the video material can be reused to find different events. This makes the approach flexible and in the case that during the analysis it becomes obvious that other log data is actually needed to carry out a study or answer a question, there is no need to collect data again.

The main contribution of this paper is to introduce a method for data collection.

- We show how we can find events automatically in mobile screen recordings to create log files. In order to validate the proposed method, we perform test runs with mobile screen recordings of a sample size that can still be analyzed manually to compare and verify the results. Another question closely related to this is what kind of events are detectable using the proposed method and how these can be defined.

- In addition, we analyze video quality requirements based on bit rate and frame rate to prepare for future large-scale, long-term studies. As high-quality video recordings easily produce big file sizes, it is an issue especially in a mobile environment, under which video quality conditions the method can still produce acceptable results.

4.2. Related Work

There are some approaches in the research community focusing on logging and analyzing human-computer interaction which are related to the proposed logging method.

The idea to use screenshots or screen recordings to get better insights has been a frequent topic in HCI research. Findings of a study exploring the possibilities of a tool for logging screenshots and interactions (in this case clicks, touches, selects etc.) show that this information helps to analyze human-computer interaction on mobile devices in a non-laboratory setup together with log data (Kawalek et al., 2008). One of the benefits of this logging method highlighted by Kawalek et al. is that the sighting of the logs combined with screenshots and information about user interactions is more time efficient compared to the review of a screencast to find usability problems.

Chang et al. (2010) presented "Sikuli Test", an approach to test desktop GUIs automatically using computer vision methods. The GUI tester first defines scripted GUI events using images and adds how to act on these events. After that, the script acts "like a robotic tester". Although the approach does not aim to generate log files from screen recordings, there are some things in common with our method, like the definition of events and detection of these using computer vision.

The project "InspectorWidget" (Frisson et al., 2016) proposes an automatic screencast annotating system using computer vision techniques. The user can (visually) program annotations and after that automatically annotate screen recordings and visualize the results or export them for further analysis. Template matching and optical character recognition (OCR) are used to recognize events. They present a test scenario for a usability check. The project focuses on desktop devices and was developed for Linux, Windows and Mac. Mobile devices like smartphones and tablet computers are not supported. Although the project's event detection is not tested or optimized to

work with mobile screen recordings, the process of detection is related to the pursued approach presented in this paper.

In addition to approaches, which tend to focus on automation, there are also several research projects that focus on the manual analysis of videos in order to investigate mobile usage behavior.

To understand end-user mobile device usage in detail and context Brown et. al collected screen recordings, audio, GPS location and application launches of iPhone users and let the participants annotate the video material in form of small diary entries for later analysis by the researchers (Brown et al., 2014). The purpose here is to investigate specific situations "framed by the concept of the 'occasioned' nature" of the use of mobile devices. Though the focus is especially to not automate the analysis of the collected screen recordings, the work shows how helpful video analysis can be for understanding the details of how users interact and use mobile devices. Based on this study McMillian et al. combined multiple sources of data to analyze mobile device usage in the field and high detail. The so-called *in vivo* method combines logs of system data like GPS position, active application and additionally video recording of the screen and the phone's camera and audio of the surrounding noise and voices (McMillan et al., 2015). They highlight the fact of differences between lab studies and real-use scenarios and the importance to analyze the multiple and combined use of smartphones.

Based on this previous related work we want to develop an approach that combines the advantages of log files and of analyzing screen recordings manually. The related approaches working with desktop screens and computer vision mention the high computational workload in general and especially of template matching. We address this in our approach by using an alternative to compare image parts in fixed regions, exploiting characteristics of mobil screen structure and by video preprocessing. In the next section we describe our approach in detail.

4.3. Detecting Events in Mobile Screen Recordings

In view of the fact that the use of computer vision methods and machine learning models for text recognition on every frame of a video causes computational effort, it is important to keep this workload low. There are several differences between analyzing mobile and desktop screen recordings which

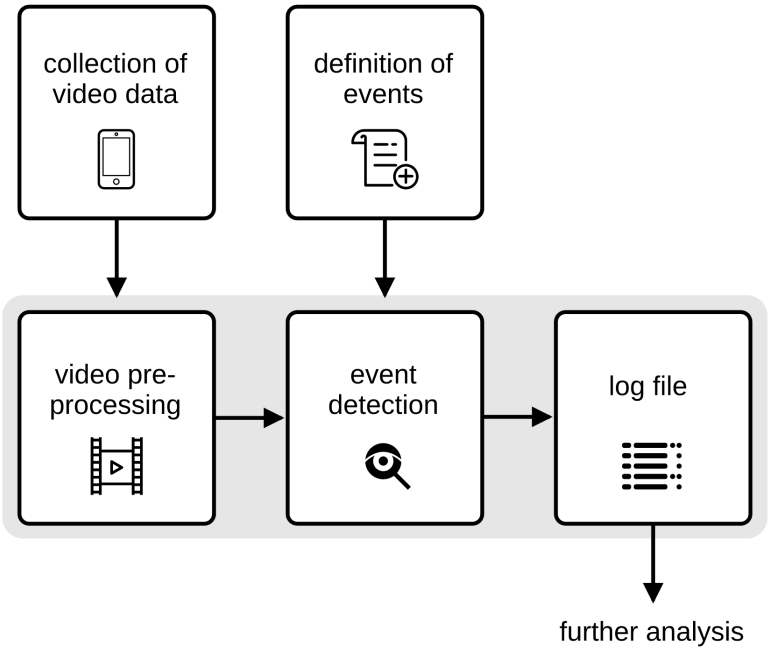

Figure 4.1.: Overview: from mobile screen recordings to log file analysis.

leads to possible optimizations for an efficient detection of events on a mobile platform. Desktop systems typically display multiple applications simultaneously in scalable and moving windows. Most mobile platforms support to display only one application at a time in full screen what reduces the cost to check where and which application or activity is present. Another major difference is that a mobile screen has a way more fixed structure. A lot of applications follow a fixed scheme. The application bar at the top or bottom e.g. usually uniquely identifies the present application and does not change. On this basis, our approach reduces the computing power required by exploiting the characteristics of the mobile platform.

4.3.1. Video Preprocessing

We interpret each video file frame by frame. As a first step, the video is slightly preprocessed. Each video frame is compared to its predecessor to calculate similarity. This speeds up the whole process a lot as there are many frames with no visible difference and therefore do not have to be processed again. The found events from the previous frame just get copied for the next log-entry. The similarity is determined by two simple and quick tests. Two con-

secutive frames are subtracted from each other, resulting in zero values for identical pixels or very small values for small changes. The maximum value and the average difference compared against a threshold are used to decide how different two frames are.

Figure 4.2.: Definition of an event: In this example two fixed areas of the UI are set (keyboard is opened and SMS chat header), a text string is fetched from an area (name of the chat partner) and we search for a small image part (smiley) in the whole frame.

4.3.2. Event Detection

Every frame is checked against a list of previously defined events. A frame can contain multiple events or no events, but the log file contains an entry for every frame. An Event contains a log message and several conditions and rules based on a GUI state. In order to define an event, a screenshot of the GUI state of interest is marked with bounding boxes representing regions of interest (illustrated in figure 5.3). We distinguish between fixed position areas and

areas that can occur anywhere on the screen. Additionally, the occurrence of text strings can be searched in areas or in the whole screen using OCR. Besides that, a text string from a screen area can be attached dynamically to the log entry.

In order to save computational time, we start by checking the event's condition that requires the lowest computational power: first comparison of fixed areas, second searching for image parts that are not at a fixed position by using template matching and afterward optical character recognition. By this order, we can exclude events early. The recognition of events works frame by frame, which makes multi-processing easy and speeds up the process considerably: we compute each frame on its own core, which facilitates the scalability of the process.

4.3.3. Comparing Fixed Areas

To compare whether a fixed area from an event definition appears at the same position in the current frame we use a perceptual hash function (Buchner, 2018). We take the hash of this fixed area from the current frame and calculate the hamming distance to the hash from the event definition to decide whether the areas are similar. Due to the already mentioned fixed structure of many mobile applications, this approach is beneficial for checking the similarity of important fixed-position areas to increase the detection speed compared to existing approaches for desktop screen recordings, where more expensive template matching techniques are used for the entire video image or parts of it. Additionally, perceptual hashes are quite robust in terms of different scaling of images or small changes and artifacts from video or image compression. We show an example in figure 5.3, in wich we check whether the keyboard is opened in the current screen. Interface elements may appear on a changing background (e.g. Android home button). Perceptual hash comparisons or template matching do not work reliably in this situation. In this case, we modified the detection process slightly. We use OpenCV's (OpenCV library, n.d.) Canny edge detection (Canny, 1986) for elements on a changing backgrounds and use perceptual hashing afterwards. By using the edged image instead of the original image it is possible to identify icons or shapes on changing backgrounds.

4.3.4. Searching for an Image Part

To search for partial interface elements or image parts that do not appear at a fixed position we use OpenCV's template matching algorithm. Searching for a part of an image in another image can be costly in computing time compared to matching fixed areas of two images. Therefore, template matching is executed in the second step, which means that many events are already excluded. An example of using this is to recognize the moving button to take an incoming call on an Android phone.

4.3.5. Searching for Text

As a final step, we use the open source library tesseract (tesseract-ocr, n.d.) for OCR to compare if a string specified in an event definition is found in a certain region of the screen or to fetch a string from a defined position on the screen to add it to the event's log message. This is helpful to distinguish a lot of events or to save the name of a chat partner for example.

4.4. Experimental Setup

In order to evaluate the method, we have carried out a series of test runs. Our approach is twofold: to begin with, we compared the found events by hand with the corresponding video material to validate the results and track down false log entries. Our intention was to identify problematic situations, events and the limits of our event detection.

Besides that, we did certain test runs with the same video material but changes in frame rate and video quality. Tabel 4.1 lists the variations of video parameters we used for exploring bit rate and frame rate limits. All video variations use the same codec, h.264. The event list and event definitions did not change during the test phase. Afterwards, we evaluated whether log entries were lost and which events were not recognized correctly or were problematic.

4.4.1. Collecting Screen Recording Data

In order to test the proposed method, we collected screen recordings on an Android phone (Google Pixel 2, Android 8 Oreo). We collected 118 recordings of a total of 68 minutes of video material during 24 hours, all in portrait mode. As these recordings may contain very personal data, we decided to use a working telephone of one of the authors to collect test material instead of external participants. The intention was to use a set of videos for testing purposes, which represents approximately a realistic average usage time per day (Böhmer et al., 2011). No special instructions or tasks were followed. These sample recordings were used for testing and validating the log results of the proposed method. At this video length, it is still possible with reasonable effort and expense to find certain events and create a "manual log file" for validating and evaluating the developed method and its setup variations. The original recording was made in high quality, full resolution (1080p) and frame rate to encode later variations of the video material in lower bit and frame rates. This makes it possible to carry out different test runs with the same material but with different video parameters in order to ensure that the results are comparable.

For collection, we developed a simple open-source Android application to record the screen (Krieter, 2018a). The application runs as a foreground service and is unnoticeable by the user, except through a notification in the notification center that informs about the recording. Battery, CPU and memory consumption are low and not noticeable on a Google Pixel 2 during the recording. The application splits the videos into usage sessions as defined by van Berkel et al. (2016) and only records if the screen is active. The application also recognizes the change of orientation between portrait and landscape and changes the video orientation accordingly. All files a stored as MP4 files and contain the start timestamp in their name.

Table 4.1.: This table specifies the originally recorded video (first row) followed by variations of the same video material but different bit and frame rate settings. The bit rate is the average bit rate.

description	frame rate (FPS)	avg. bit rate (KBit/s)	file size (MB)
original quality	20 (avg.)	4418	2191
10 FPS, low quality	10 (fixed)	227	113
5 FPS, low quality	5 (fixed)	203	101
5 FPS, very low quality	5 (fixed)	51	25

The originally recorded video material has a variable frame rate according to changes and light situation of 20 FPS on average. By lowering the bit and frame rate we reduced the file size from originally 2191 MB to 25 MB. The indicated bit rate is the average bitrate of all 118 videos of one video quality (Table 4.1).

4.4.2. Eventlist for Testing

Table 4.2.: This table lists general UI events that can occur in parallel with other events.

#	Event description	perceptual hashes	edege detection	template matching	OCR
27	keyboard is opened	✓			
28	smiley keyboard is opened	✓			
29	the ”k” key is pressed	✓			
30	the home button is pressed	✓	✓		

Table 4.3.: This table shows app-specific or task-specific events that are used for test runs and which techniques the individual events use, marked by checkmarks. The "x" indicates that a text or numeric variable from the video material is attached to the log entry.

#	Event description	perceptual hashes	edege detection	template matching	OCR
1	SMS chat opened with chat partner x	✓			✓
2	SMS message/chatlist opened	✓			
3	SMS starting a new conversation screen opened	✓			✓
4	Whatsapp, chat list opened	✓			
5	Whatsapp, chat list opened, the list is scrolled down	✓			
6	Whatsapp, number of chats with new messages x	✓			✓
7	Whatsapp, chat opened, chat partner x	✓			✓
8	Whatsapp, using rear camera in a chat	✓			
9	Whatsapp, button for taking a picture pressed	✓			
10	Instagram opened	✓			
11	Instagram feed opened	✓			
12	Instagram feed opened, there are new unread messages	✓			
13	Instagram direct chat opened with chat partner x	✓			✓
14	Pinterest opened, chat with chat partner x	✓			✓
15	BBC news opened, viewing top stories	✓			
16	home screen present	✓			
17	Android all apps menu present	✓			
18	calculator opened	✓	✓		
19	contacts opened, viewing list	✓			✓
20	contacts opened, editing a contact	✓			
21	calling, trying to call contact x	✓			✓
22	calling, in call with contact x	✓			✓
23	calling, hanging up contact x	✓			✓
24	lock screen, incoming call from contact x			✓	✓
25	lockscreen present	✓	✓		
26	lockscreen present, there are notifications	✓	✓	✓	

The exact definition of events is critical for reliable log results. We defined the test events based on Android screenshots, as our screen recordings for evaluation come from an Android device, but by adjusting the event definitions, the approach would also work for recordings from other devices (e.g. iPhones). When defining an event, the following question is decisive: Which features make the event or action differentiable in the GUI state? We defined a list of 30 sample events representing a series of user interface (UI) events and specific situations when using certain applications. The intention is to test a series of events that present a number of difficulties for the detection: Distinguishability from other similar events, or no events at all, and very short or fine-grained events that could be difficult to capture if the frame rate and quality of the collected video material are low. The events vary and range from very short, fast interactions such as triggering the camera in a Whatsapp-chat to more general interactions such as the presence of the keyboard or smiley keyboard. By "short" events we mean fast interactions of less than one second, such as pressing the "k" key on the keyboard or using the Android home button. Another example is to register the opening of a chat in several chat applications and attaching the names of the chat partners to the log entries (similar to figure 5.3). A complete list of all events can be found in Table 4.3 and 4.2. The variable "x" stands for a value that is retrieved from the event screen, e.g. the name of an incoming caller or the number of new chats in the Whatsapp chat application. Besides that, the tables indicate which techniques the event definitions utilize. The UI events in Table 4.2 are slightly different to the app-specific events, as they can occur in parallel with other events. E.g. the home button can be pressed in any application or the keyboard is used concurrently with many other events in applications.

4.5. Results

Using the original video material, our approach produced a log file containing one or more entries for all 79790 frames. We transferred this frame-by-frame log file into an a different format to facilitate analysis. In this final log format an event is defined as a contiguous group of the same results for consecutive frames. E.g. if the event "keyboard opened" was found in the last 100 consecutive frames, we take this as an event with a starting time and of a certain length of time. This results in a list of log entries in which each entry rep-

resents an event with a starting point and a duration in chronological order, what makes comparisons with the corresponding video material easier. In addition, we have stored each frame that contained an event together with the log message to make validation of the results more feasible. The processing of the original video material on an Intel i7 CPU took about 10 hours, i. e. an average of 133 frames per minute.

Our approach did not miss any events when processing the originally recorded high-quality videos but produced 2790 log entries with faulty log messages. Most of these entries occurred from OCR when used on frames from a screen transition when switching or closing an application. Taking a closer look at the frame-per-frame log files shows that in some cases of switching applications problems occur with detecting certain events. In these situations, we still find an open chat application with a certain person for example, but when OCR tries to get the name from the chat header, the started transition into another GUI state results in a messed up text string. This produced faulty log messages for events that utilize OCR to retrieve text from frames. Almost all other faulty interpreted frames resulted from the "calculator opened" event which was based on comparing perceptual hash values of the lower screen area with edge detection applied to the image before which led to the unexpected effect that other frames got interpreted randomly as calculator screens. All faulty frames together resulted in a total of 17 (1.3%) false positive events. As a false positive we consider the finding of an event which contains a correct log message, but which did not appear at this position in the video. The calculator application was present one time in one of the 118 test videos but was found 12 times. Besides that, the number of new messages in chats in Whatsapp was recognized (as 1 and 3) although there were no new messages. The remaining false positives are distributed between the events "Instagram, feed opened" and "SMS chat opened". In the latter case, some frames were interpreted as SMS chat, although they were screens for starting a new SMS conversation.

All other events were detected as intended and are part of a detailed log file of 931 entries (without faulty OCR log messages and false positives) representing a detailed overview of phone usage and in-app events. In total, 40 unique events occurred in the final log file. Short or very temporary events like touching the home button (40 times) or using the k-key on the keyboard(24 times) were detected reliably. The correctly recognized chat events led to log entries of chats with 11 different people in 4 applications (Whatsapp, SMS, Instagram direct message and Pinterest). The smartphones rear camera was used two

times inside of Whatsapp to take pictures and send them to two different people. Phone call events were recognized correctly (one outgoing and one incoming call to the same person).

As problematic we found using OCR in GUI transition states leading to unpredictable text results. All these events are conditioned by previous checks of perceptual hashes of fixed areas of the frame to identify the event which are more robust to changes: when comparing perceptual hashes, a transition frame may still contain a Whatsapp header, but OCR no longer delivers stable results. Another problem encountered in the results is that edge detection does not work reliably in all situations and leads to several false positives.

4.5.1. Phone and App Usage Sessions

To graphically display the log results in a descriptive way we visualized the log file as phone usage sessions containing application usage sessions in reference to van Berkel et al. (2016). An example of a short part from the log files is shown in Figure 4.3 respresenting one of the 118 recorded phone usage sessions. The phone usage session, in this case, has a duration of 128 seconds and contains the use of two different applications and the presence of the Android home screen. The application session illustrates the detailed use of in-app events and additionally parallel UI events (table 4.2), like the opened keyboard in a second line. In this case, the use of the chat application Whatsapp with two different persons and viewing and scrolling down the list of chats is shown. The application session in this example has a duration of 28 seconds.

4.5.2. Results with Lower Video Qualities

Following the test run with the original video material, we carried out further tests with the same material, but in poorer quality and at a lower frame rate as described in Table 4.1. This led to differences in the resulting log files and in missing the occurrence of events.

Table 4.4 shows how many events were missed and the false positive rate per video variation. Lowering the quality and frame rate did not necessarily result in a higher false positive rate, which stayed low between 0.7% and 2.0% for all video qualities. The undetected events are not distributed evenly over the entire event list. Some events performed significantly worse, others

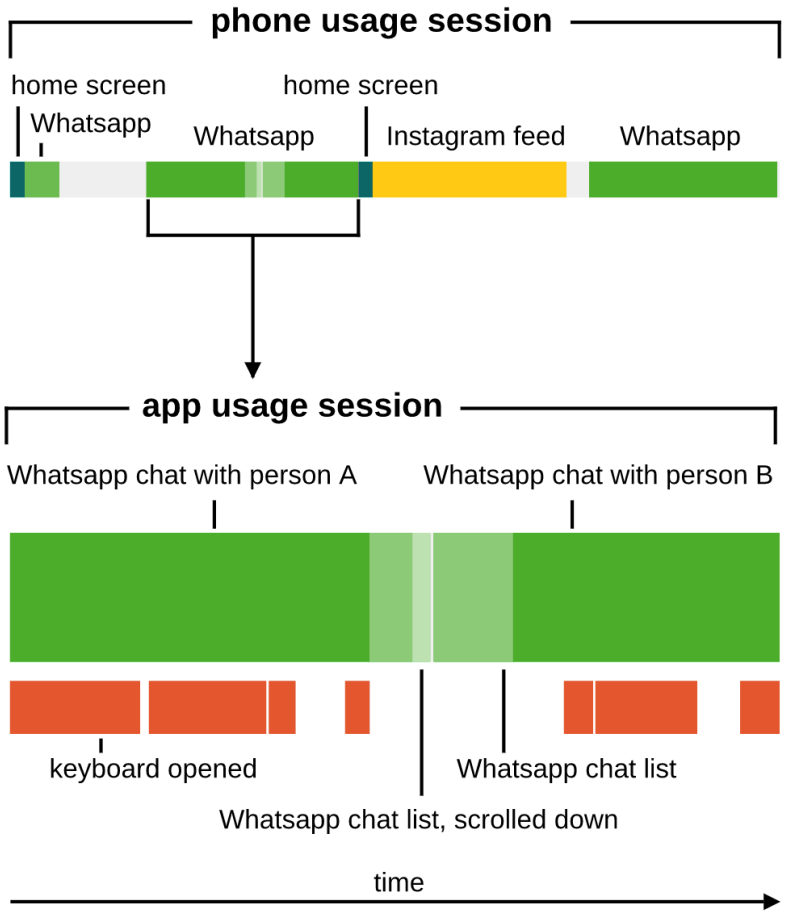

Figure 4.3.: A part of the log results visualized of in form of a phone usage session and application usage session.

remained at a high level. The findings from the results of the original video occur in the lower quality version as well. Most false positives resulted from misinterpreting frames as calculator frames, too, exactly as for the original video material. The same is true for using edge detection on image parts before comparing their perceptual hash values, although this negative effect is much stronger for some events. The home button event, for example, was not missed in the original material, although the event definition uses edge detection as the button can occur on different backgrounds. But all log files of the three lower quality versions of the video material show significant loss of the number of found home button events. The test run using the lowest bit rate videos resulted in zero home button events.

Figure 4.4 shows the performance of some selected events per video quality. We have selected a number of events as examples, which are stable, which are

Table 4.4.: This table lists the results for missed events and false positive rate of the test runs with low quality video material.

video material	missed events	false positives
10 FPS, low quality	18.2%	1.0%
5 FPS, low quality	24.8%	0.7%
5 FPS, very low quality	30.4%	2.0%

for the most part stable, and events that have produced very poor results. The diagram indicates the proximity of the number of found events in relation to the number of events found in the original video material in percent. This is not the same number as the number of missed events. The expectation is that despite the reduction of video quality and frame rate, the same log file will be created. However, it is possible that the number for an event may vary without missing it. For example, the Instagram direct message chat event was found 2 times in the original material, but 7 times in the lowest quality material. The second occurrence of the event is approx. 5 seconds long in the original videos, but is divided into 6 very short events, which add up to approx. 5 seconds when using the 5 FPS videos with a very low bit rate to generate the log file. Some frames within the event are wrongly recognized due to quality losses. This means that the event was generally recognized, but the number of events in the log file is not as expected.

On average for the complete event list, the difference to the number of found events in the original video material is 14% for the 10 FPS material, 19% for the 5 FPS videos and 33% for the 5 FPS videos with very low bit rate. The averages were derived from the results for all 40 unique log file events. Many events have stable results in all three test runs, but some of them are very different from the results with the original video material. The selected events in figure 4.4 show exemplary how single events scored. The first two events (Instagram, incoming call) did not show any differences compared to the original log file, this was true for 8 events in total. The next two events (smiley keyboard, k-key pressed) are slightly different, but still over 80%, which holds true for a group of 14 events. When the k-key is pressed, the animation lasts about 0.3 seconds, long enough to be recorded at a frame rate of 5 FPS. The last three events are examples of events that were very far from the results of the original video material in at least one case. The lock screen event only reached 80% for the 10 FPS videos and had a difference of more than 60% for the lowest quality videos. The last two events did not occur at all in the log files from the video material in lowest quality. The home button event is

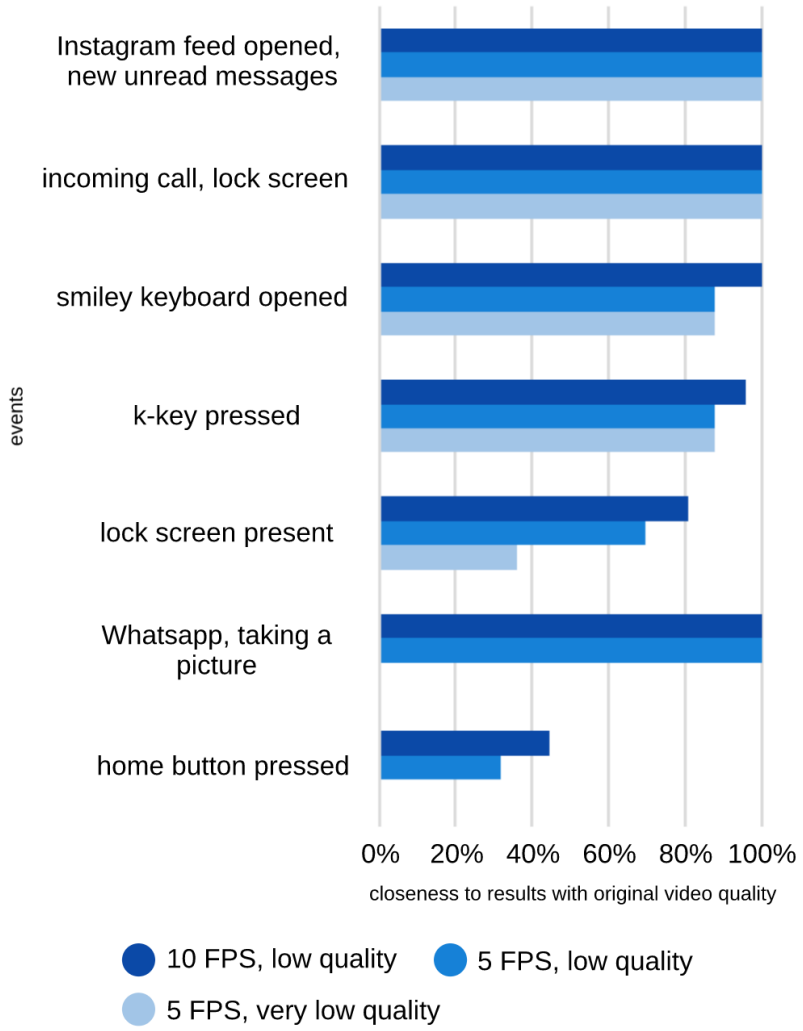

Figure 4.4.: This diagram shows how close the results for different video qualities are to the results with the original material in percent. For example, for the "home button pressed" event only a little more than 40% of events are found at 10FPS, at 5FPS about 30% and at 5FPS with very low video quality 0%.

based on only a part of the button animation whose presence is too short to be reliably captured at low frame rates and bit rates.

4.5.3. Video Preprocessing Results

Table 4.5 lists the number of skipped frames by comparing the similarity of consecutive frames during preprocessing the videos before the event detec-

tion process starts. All test runs have benefited considerably from video preprocessing. Running through the original and low quality 10 FPS and 5 FPS material more than half of the frames were skipped. We expected a higher percentage of redundant frames at higher frame rates, which is the case for 10 FPS and 5 FPS, but not for the original video material at an average of 20 FPS. This might result from the adaptive frame rate of the original screen recordings which produces less redundant frames. Using the lowest quality resulted in skipping slightly less than half of the frames. The difference of 10% between the two 5 FPS versions might result from apparent differences in the stronger compression of video frames and related artifacts. This preprocessing step saved a considerable amount of computational time, as checking each frame against the event list is the most costly step in the whole process.

Table 4.5.: This table shows how many frames are skipped by preprocessing the video material.

video material	skipped frames
original quality	54 %
10 FPS, low quality	64%
5 FPS, low quality	56%
5 FPS, very low quality	46%

4.6. Discussion

Most previous work on mobile applications usage that works with log files relies on log events that do not provide insights about what users do inside applications, limiting research in this direction. We present an approach that is capable of providing log files in a different level of detail about what users do in their applications. It is not necessary to implement log commands in the source code of any application beforehand. This makes the approach flexible and the definition of log events can be done after collecting video data.

Our approach combines two methods of different characteristics. Log file generation in general, is an exact long-term instrument to document events in computer systems. Analysing interactions between users and applications by manually viewing screen recordings is a short-term method. Using computer vision and machine learning methods to create log files is less exact in its outcomes as these techniques rely on probabilities instead of exact meth-

ods. Although finding an event like using a certain chat application in mobile screen recordings is different to finding scenes containing a male lion in a nature documentary for example. Mobile screens have a fixed structure and therefore make the use of rather simple computer vision methods like template matching or perceptual hashes possible, in contrast to more advanced methods that are necessary to classify a lion in wildlife videos. Current related work uses manual sighting to analyze video material or template matching and OCR on videos from the desktop platform. By exploiting the fixed structure of mobile applications our approach saves computation time compared to desktop approaches. The determination of the similarity of certain fixed image areas using perceptual hashes is sufficient to identify events. Our approach struggles when detecting UI elements on changing backgrounds. We use edge detection to extract the foreground, but how we use this does not produce reliable results in our evaluation. Template matching was barely used in our event list as most areas of interest of our example events were at fixed positions. This stresses the difference between mobile and desktop UIs. In our approach template matching produces better results with the high-quality videos compared to our test runs with low video quality. Using OCR on transition frames from switching between GUI states are causing the largest amount of corrupted log messages. This needs to be addressed from a perspective of how these situations can be identified to prevent using OCR on these frames as tesseract delivers accurate results when used on clear video images.

Our test runs indicate that the proposed method can deliver detailed log files. For long-term use on mobile devices the file size of video data should be low. For this reason, we test different video qualities with smaller file sizes. Missing an event does not necessarily correlate with a low frame rate or bit rate. However lowering the framerate lowers the number of occurrences of short events on video frames and therefore the chance of detecting them. Comparing the results for very short events like pressing the k-key shows, that even with low quality and frame rate a detection is possible. Besides keeping the file size low on the mobile site it is necessary to optimize the recognition process for using our approach at larger scale. The video preprocessing showed that all tested video variations contained a lot of frames that are redundant for our recognition. This leads to the question of whether this step should be implemented earlier: already when recording the screen of a mobile device.

4.6.1. Privacy

In contrast to very transparent self-reporting methods or logging of more general mobile system events, screen recordings or screenshots can contain very sensitive and personal information. Our approach possibly analysis everything that occurs on the user's screen. As a side-effect, this implies the need to develop an effective privacy concept that ensures that users can stay in control of which events they share with who. Following the Privacy by Design (Cavoukian, 2011) approach, this needs to be addressed right from the beginning. Although in the current stage of development the room for abuse scenarios is small, it is important to take this into account and to make sure that participants screen data stays private in further studies.

4.6.2. Limitations

Our log files show that videos of 5 FPS are sufficient for a range of our event definitions and can work for short interactions. But to use this bit and frame rate in a large scale study, more test runs and further development are needed. Besides that, we use video recordings from using the phone in portrait mode. Changing the video orientation to landscape would not work with the event definition list used for our test runs. We worked with absolute numbers defining the bounding boxes for events. For landscape videos the same event list would have to be checked and adjusted accordingly to the changed GUI and aspect ratio. Working with relative values instead might solve this to some degree. One strength and weakness of our approach is that almost everything that happens on the screen can be tracked. However, events that can be triggered by the user via hardware buttons, such as the volume or skipping of a music track, cannot be detected.

4.7. Conclusions

In this paper, we present a concept, implementation and evaluation of a method to analyze mobile application usage by automatically generating log files based on mobile screen recordings. We showed that exploiting charac-

teristics of mobile screen recordings is possible. The following list sums up our key contributions.

- The proposed logging approach can provide fine-grained log files on in-app events and general interactions independent of modifying the source code of any application.
- In contrast to similar desktop approaches it is possible to exploit the structured GUI of a mobile device: many elements appear at fixed positions of the screen. We compare the similarity of fixed areas of the screen with fixed areas of the event definition to successfully identify a wide range of events. Our video preprocessing eliminates redundant frames and thereby speeds up the recognition process.
- We show that using perceptual hashes, template matching and OCR are appropriate to identify events in mobile screens to create log files. Recognizing elements on changing backgrounds using edge detection does not work reliably in our approach.
- We compare the log results of different video qualities. A wide range of events is detectable successfully even in low bit and frame rate videos. Resulting log files are missing several of our example events depending on the specific event and its definition.

The source code and event list of this project is available open source (Krieter, 2018b).

5. Privacy and Scalability

This chapter contains the publication with the title "Can I record your screen? Mobile screen recordings as a long-term data source for user studies" from the proceedings of the 18th International Conference on Mobile and Ubiquitous Multimedia (MUM'19) published by the ACM.

5.1. Introduction

Tracking mobile application usage is essential to understand and improve mobile human-computer interaction (HCI). Mobile screen recordings are a highly useful data source in HCI: by observing, we can follow almost everything a user is doing when interacting with a device (Tang et al., 2006; Kushniruk & Patel, 2004; Haak et al., 2003; Barnum, 2010; Lazar et al., 2017; Aho, Kanstren, et al., 2014; Vuong et al., 2017). The manual analysis of screen recordings is usually applied for short periods, as viewing the video material is time-consuming. There are recent approaches automating the analysis by generating log files based on screen recordings (Frisson et al., 2016; Krieter & Breiter, 2018a; Reeves et al., 2019). Through this, logging capabilities can be extended from rather general system events to detailed in-application events by using methods from computer vision and machine learning to analyze the screen output of mobile devices (Krieter & Breiter, 2018a; Reeves et al., 2019; Vuong et al., 2017). In comparison, most mobile log files provide rather general device usage events e.g. which application is active (McMillan et al., 2015; Böhmer et al., 2011). Screen recordings can cover situations, that can not be covered with 'classic' logging approaches, which rely on implementing log commands in applications.

The downside of using screen recordings as a data source in user studies is, that recording the screen of a user permanently for later analysis poses a major threat to the personal privacy of participants in research studies (Tang et al., 2006; Reeves et al., 2019; Valderrama Bahamóndez et al., 2014; Vuong

et al., 2017). Most recordings from real-world mobile device use contain personal and highly sensitive information. According to previous research using screen recordings, the main reason for users declining to participate is privacy concerns (Reeves et al., 2019; Tang et al., 2006). This results in a typically very rich and resource-intense data collection, limited to a low number of participants (Yeykelis et al., 2018). The screen recordings are collected on the user's device and sent to a server for processing and log file generation, ignoring the sensitive nature of this data. Previous research has shown that data stored centrally with third parties is at conflict with user privacy demands (Spiekermann & Cranor, 2009; Barkuus & Dey, 2003; Hong et al., 2003). In addition, the central processing of videos needs high computational resources (Krieter & Breiter, 2018a; Frisson et al., 2016). Additional research in privacy issues related to these methods is needed to enable further and larger research samples using this data source (Reeves et al., 2019; Valderrama Bahamóndez et al., 2014).

In this paper, we address these two major issues, when using screen recordings as a long-term data source in HCI: privacy concerns of potential participants and scalability due to a central cloud structure. Following the Privacy by Design (PbD) principle (Cavoukian, 2011) we change the process of creating log files to protect users' privacy and increase users' willingness to participate in studies using this method as a data source. Additionally, we increase the scalability of the method by aiming at applying this method under real-world conditions and at a large-scale on mobile devices. We decentralized the server-based automated video analysis to the mobile device to create log files directly on the device. As a consequence, the user's highly sensitive screen recordings do not leave the user's mobile device. The resulting log files are then anonymized and finally sent to a server for further analysis. Besides the privacy advantages, the decentralized structure eliminates the requirement of high central computational resources for the log file generation on the server-side and saves bandwidth on the client-side, as the screen recordings do not have to be transferred.

Our main contribution is to demonstrate how we can use mobile screen recordings as a long-term data source for log files in a scalable and privacy-friendly way to analyze the use of mobile applications in detail.

- We present an Android implementation of a logging approach on mobile devices to find and track events based on the visual screen output of the device while respecting the privacy of the user. Based on PbD we develop and implement a privacy concept to increase the availability of users for

participation in studies applying our, or similar, methods. To evaluate our privacy concept, we conducted a survey comparing our approach to two other logging approaches on mobile devices.

- We conduct a two-week pilot study on a mobile device to explore how the approach performs under real-world and long-term (for screen recordings) conditions and to demonstrate what kind of data we can collect. To check how our approach runs on different operating system versions and devices we perform test runs on ten different Android setups.

The logging application presented in this paper is available open-source to enable other researchers to generate their own data and make our approach transparent and reproducible results (Krieter, 2018c).

5.2. Related Work

Using screen recordings as a data source has been a common subject in HCI research. Both manual and automated analysis is applied. In both cases, privacy concerns and legal restrictions accompany the process, especially when conducting studies in non-laboratory settings.

Tang et al. explore the use of screen-recording as a method of gathering data in field studies (Tang et al., 2006). They note that it is not easy to find participants for these studies due to privacy considerations, as screen recordings contain all interactions of the user with the system. Participants are aware that they share private communication for example. They mention that rich empirical data collection such as screen recordings requires a high level of trust between participants and the research team. Besides that, it is important to give users control over stopping and starting the recording, and limiting access to the video material. Also, approval for presenting anonymized video material to third parties is appropriate, to ensure participants are aware of what they share.

For a study by Vuong et al. (2017), the computer screen of ten participants is recorded for 14 days to investigate how digital task recognition can be achieved using unsupervised topic modeling. Out of 14 participants, one left the study for privacy reasons. They discuss privacy issues related to monitoring screen activity and suggest a human-centered design that allows users to maintain control over their data by processing data locally.

Reeves et al. (2019) present a framework for collecting and analyzing sequences of screenshots they call 'screenome'. They collect screenshots on mobile and desktop devices at five-second intervals, transfer these to a server and extract text and images and organize the results in a searchable database. They state that the main reason for potential participants to decline to join their research studies were privacy concerns due to the invasive nature of collecting and analyzing screen data. The privacy of text messages is particularly mentioned as a problem. In terms of lowering the risk for participants, they suggest "performing local analysis and only transferring summary results".

Brown et al. use mobile screen recordings to manually analyze and understand how users interact with and use their mobile devices (Brown et al., 2014). Besides recording, they utilized GPS locations, logs of application launches and audio recordings of iPhone users as qualitative data sources. To protect the privacy of their participants, they let the user check and annotate their screen recordings in a web interface before making them available to the researchers. Through this step in the data collection process, they give their participants the option to decide which screen recording they want to share and which they want to keep private.

A related approach by Krieter and Breiter on the mobile platforms aims at analyzing mobile application usage in detail by generating log files based on mobile screen output (Krieter & Breiter, 2018a). Utilizing existing computer vision and machine learning methods they automatically search for events in mobile screen recordings to create log files. As a first step, the screen of a mobile device is recorded and afterward the videos are transferred to a server for the detection of events and the creation of log files. In terms of privacy, it is very problematic to use this method on private mobile devices. The screen is recorded whenever it is turned on and therefore might contain very personal details or copyrighted content. Although only certain events are being searched and written to the resulting log files, this sensitive video material still has to be transferred to the researchers' server and possibly can be viewed. They also state that local processing could help improve the situation for participants.

Following this, we use the PbD approach as an orientation for designing our logging method. The concept of PbD emerged in the '90s and contains seven principles to take into account: proactive not reactive, privacy as the default setting, privacy embedded into the design, full functionality, end-to-end security, visibility and transparency, respect for user privacy (Cavoukian, 2011). The idea is to make PbD a standard, to improve privacy by consider-

ing it before and while developing a product (Schaar, 2010). Rubenstein and Good notice that although companies and regulators agree on PbD as an important aspect to incorporate, they struggle on how this can be achieved in detail (Rubenstein & Good, 2013). There are more indications pointing in this direction. Ayalon et al. conducted a study on how developers make decisions about privacy, to investigate why approaches like PbD are still not a common principle in software development departments(Ayalon et al., 2017).

Closely related to this is the 'privacy-by-architecture' approach that is defined by trying to implement the minimization of the collection of personal data that can be identified and focuses on its anonymization. In addition, the processing and storing of personal data is strongly preferred on the client-side instead of network storage (Spiekermann & Cranor, 2009). Spieckerman and Cranor advise this approach over the 'privacy-by-policy' approach that emphasizes the notice and choice for users on which of their personal data is used but does not implement privacy by design or default.

5.3. Privacy Concept

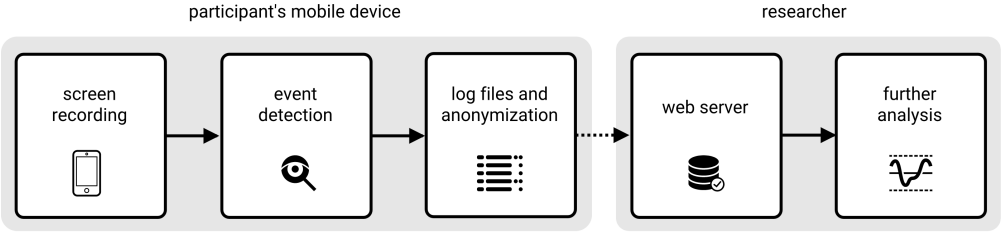

Figure 5.1.: Overview of our process: from screen recordings to privacy-friendly log files and analysis.

The two closely-related logging approaches presented are focusing the technical realization on generating log files in difficult contexts, but do not take into account how privacy or anonymization of the resulting log entries can be achieved to make recruitment of participants easier (Reeves et al., 2019; Krieter & Breiter, 2018a). We derive several privacy issues related to using screen recordings from the literature and take them into account in our concept. Following Spiekermann & Cranor (2009) we develop a privacy concept guided by the PbD principle that implements privacy in its architecture. Technically

related to the approach of Krieter & Breiter (2018a) we implement a logging application and process that takes privacy into account from the beginning. In this section, we describe how we have dealt with each point of the Privacy by Design Principle (summarizing related aspects).

5.3.1. Proactive not reactive

By automating the analysis, we can be very selective about what we extract from the screen videos. Manual analysis is very different from that because a person will always see and interpret more than just the desired data from the recordings. Automated analysis of screen recordings can thus strengthen the privacy of the participants. By remodeling and implementing the process of creating log files from screen recordings, we try to prevent privacy risks before they occur and make the method easier to use in studies. Instead of having participants delete videos they do not want to share after data collection, the research team never has access to the screen recordings (see figure 7.1).

5.3.2. Privacy embedded into design and as the default

The most important step of our privacy concept is to keep the screen recordings on the user's mobile device instead of uploading them to a central server for processing. The recordings may contain personal and private data, from personal images to passwords, end-to-end encrypted text chats or online banking information, to name some examples. For this reason, it is unlikely to find participants for user studies involving long-term screen recordings in a real-world context (Tang et al., 2006; Reeves et al., 2019). Instead of performing the detection of events centralized on a server, we reimplemented and optimized the detection and log file generation for mobile devices (figure 7.1). Through this step, the sensitive video material can be deleted right after analysis and log file generation. A similar research approach is followed by Boker et al. (2015).

5.3.3. Full functionality and security

The intention of our method is not to interfere with the user's experience and functionality with the mobile device and to provide high-quality anonymized data on the research side. We added a parameter to the event definitions to determine if this event's log message might contain private or personal data. An example is an event that saves the name of a chat partner (figure 5.3). In this case, we anonymize the name by creating the MD5 hash of the name and a random secret string. For anonymization MD5 hashes are still sufficient (Thompson, 2005). When the app is installed the first time, it generates a random pseudonym for this device and a secret string. The pseudonym is used by the server to store the received log data per device, without being able to directly tell which user is sending the data. The random secret string is added to all text that gets anonymized through hashing, to prevent the success of a hashtable attack. Without the secret and if the context is known, e.g. that the hash contains the name of a person, it would be easy to de-anonymize the name with a hash table attack. The log data is transferred via a secure connection.

5.3.4. Tranparency and respect for user privacy

The user gets informed when she or he starts the application about the purpose and function of the ScreenLogger application (figure 5.2). The user has the option to stop the analysis of the recordings of the screen at any time. Simple control and basic notifications are preferred to control privacy and access to personal data (Hong & Landay, 2004). When the recording is active, there is a permanent notification in the notification center informing the user about the running background service. Besides that, the operating system always shows an icon in the status bar to signal that the screen is being recorded. To remind the user what the application is scanning for, a list of all event descriptions is available (figure 5.2). The privacy of the user in the resulting log file is still very dependent on the events that are searched, as the events can be defined quite freely based on any GUI state. Therefore, it is important to inform about these events. In this case strong 'privacy-by-policy' enforcement is needed (Spiekermann & Cranor, 2009).

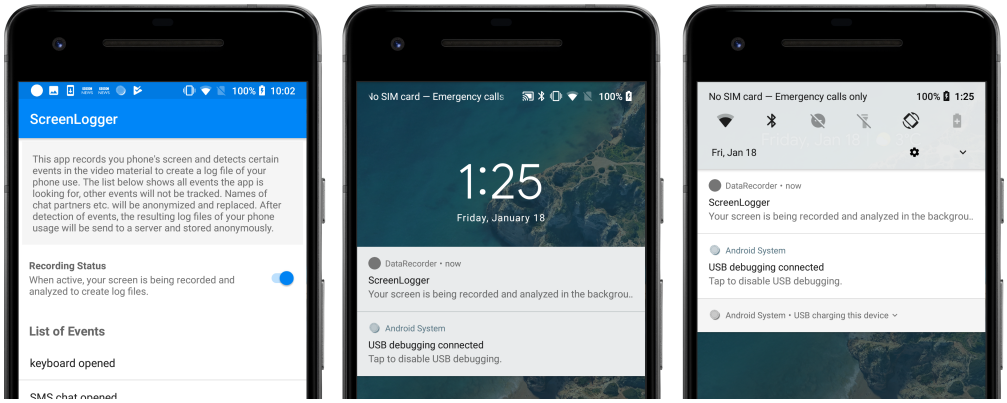

Figure 5.2.: The left screenshot shows the UI of the ScreenLogger application, which gives control over the data collection and informs the user. The screenshots in the middle and on the right show the permanent notification in locked and unlocked screen state.

5.4. Technical Implementation

Our approach uses methods from computer vision and optical character recognition (OCR) to search for certain graphical user interface (GUI) states in screen recordings and save the results as a log file. Technically the app is orientated towards the implementation of Krieter & Breiter (2018a). As one part of our privacy concept, we decentralized the detection of events from the server-side to the user's mobile device. This implies some changes and adjustments due to the limited resources of mobile platforms compared to desktop or server systems. Figure 7.1 gives an overview of how the "ScreenLogger" application processes the screen videos to create log files, anonymizes these and transfers them to a server for further analysis. Our application is written for Android devices. An advantage of our privacy concept is that the decentralization of the event detection also eliminates the need to provide high computational power on the server-side to analyze hours of screen recordings of multiple devices.

5.4.1. Collecting Video Data

As a first step, the screen of the device is recorded in the background. The recording starts as soon as the display is turned on and stops when the screen is turned off by the user or power saving. Each video file is stored with a times-

tamp on the device. We record in full-screen resolution and high video quality. The frame rate is adjusted by the device, depending on the light situation. In our test data set of screen recordings (for debugging), this leads to an average of 20 frames per second (FPS) and an average bit rate of 4418 KBit/s (Krieter & Breiter, 2018a). The video quality influences the correctness of the log results (Krieter & Breiter, 2018a). High quality leads to better results of the applied computer vision and OCR methods. Since the screen recordings are only stored temporarily and are not transmitted via the Internet, the large size of the video files is not particularly relevant. For that reason, we use the same video quality settings in our Android implementation as in the test data set.

5.4.2. Defining Events

We search every video frame for a list of events, resulting in a log file that contains an entry for every frame. These events are defined as a GUI state we are interested in and a log message. An event definition can contain several conditions and is based on a screenshot. An example is given in figure 5.3. Fixed areas describe an area of the screen that is compared for the similarity of the same area of every video frame. This helps to speed up the detection process significantly, compared to searching in the whole video frame. Due to the often fixed structure of mobile user interfaces (in contrast to desktop operating systems), it is possible to search for an element at specific locations (Krieter & Breiter, 2018a). The keyboard for example always appears at the bottom of the screen, or an SMS chat header always appears at the top of the screen. For image parts that can occur anywhere on the screen, we use template matching. Additionally, we search for text using OCR. The example event definition we used for our survey study is shown in figure 5.3. For more examples of what kind of events we used and what data we can generate see the second part of the results section.

5.4.3. Finding Events in Mobile Screen Recordings

To determine the similarity of fixed areas from the event definitions and a video frame we use perceptual hashes. We take the hash of this image part from an event definition and calculate the hamming distance of the hash of the same image area of the video frame. Unlike the implementation of (Kri-

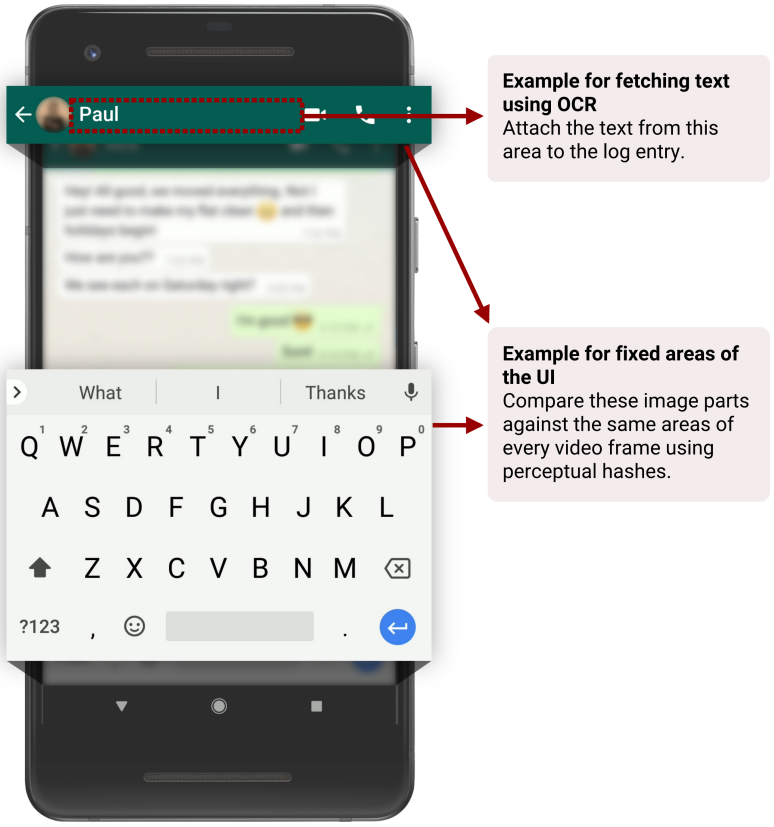

Figure 5.3.: Illustration of the example event we used in our test scenario for our user survey. We use the upper and lower screen area for perceptual hash comparisons to identify when a WhatsApp chat is opened and use OCR to get the name of the chat partner. Template matching is not used in this example.

eter & Breiter, 2018a), we use an average hash algorithm to calculate the hash values. The original implementation uses a more complex perceptual hash algorithm utilizing discrete cosine transformation (DCT), which is significantly slower, but shows better results when it comes to small changes of the image by video compression for example Zauner (2010). We switched to the average or mean hashes, due to the more limited resources on mobile devices compared to server systems. By using high video quality without strong artifacts through compression, we see comparable results for most events. Almost all of our test events contain hash comparisons, as this results in quick exclusions of possible events. OCR and template matching need considerably more computational time and are therefore only executed if the hash distance of certain areas is below a threshold. Similar to the server-based implementation we use the tesseract library (tesseract-ocr, n.d.) for Android to find text or fetch strings from frames and attach them to log entries. To speed up OCR, an event defi-

nition contains the position where to look for a text string in the video frame. To search for image parts in the video frames we utilize OpenCV (OpenCV library, n.d.) and its template matching algorithm. During the development, we benchmarked our implementation compared to the open-source implementation(Krieter & Breiter, 2018a) with a test data set of 118 screen recordings and 30 test events and achieved slightly less accurate log files (deviation of 6.8% on average). Only some rather short fleeting events were recognized less precisely, because of switching to average hash functions. The event for pressing the home button resulted in only beeing detected in 43% of the cases, for example. The detection of events runs in a background service that also records the screen. The event detection on the videos is only executed if the screen is turned off and the phone is plugged in. Otherwise, chances are high that at some point the process gets killed by the OS, as it is consuming computing, memory and energy resources constantly while generating log files. The idea is, that during the night, when most people charge their phone and have a wifi connection, the video material of the day can be analyzed. After processing a video, the recording gets deleted and the log file for this video is sent to a server via an SSL encrypted connection if an internet connection is available.

5.5. Experimental Setup

Our research design is two-fold. On the one hand, we evaluate our privacy concept in an online survey and compare our method with two other scenarios of data collection. On the other hand, we collect data for two weeks in a pilot study on one device and test our application on ten different Android setups to explore how the approach performs under real-world and long-term (for screen recording) conditions on several devices and OS versions. To conduct our survey, we have constructed an exemplary HCI research scenario suitable for our methodology. The scenario is to collect data for studying chat behavior in the popular chat application WhatsApp for a week. The data we collect is how long a chat is opened, the name of the chat partner (anonymized) and if the keyboard is opened. We choose this research scenario because it is privacy-invasive as we collect data about private chats. Additionally, we cannot collect this data, in the same way, using other 'standard' logging methods as WhatsApp is a closed application and the chats are encrypted. As mentioned in the

related work section, tracking of messaging is especially a privacy problem for potential participants (Reeves et al., 2019).

5.5.1. Survey on privacy concept

We ask the participants to imagine being asked to join a study that collects certain usage data on their smartphones and to rate different ways of data collection. The survey we setup contains an introduction explaining our HCI research scenario and four questions. Answers are set up as a four-point Likert scale: strongly disagree, disagree, agree, strongly agree. With the first question, we ask for the participants' willingness, in general, to participate in scientific studies. After that, we present three different logging scenarios and ask whether the respondent would participate in the study described if this scenario was applied. The method presented in this paper is described as 'Scenario 2'. Scenario 3 represents the server-side approach to generating log files from screen recordings comparable to Krieter & Breiter (2018a) or Reeves et al. (2019). The first scenario can be seen as a reference example that would work without screen recordings.

- **Scenario 1:** Log files are used to record (1) when a chat was active, (2) how long a chat was active in the app and (3) which chat was active. The collection of data is done in the background and anonymized. The anonymized log files are transferred to a research teams's server and then analyzed (screen is not being recorded).

- **Scenario 2:** Your screen is permanently recorded in the background. The collected screen recordings remain on your smartphone and are automatically searched for all WhatsApp chats. After that, the screen recordings will be deleted directly. Anonymous log files are created about (1) how long, (2) when and (3) which chat was active. These log files are transferred to the research team's server. The research team never has direct access to your screen recordings.

- **Scenario 3:** Your screen is permanently recorded in the background. The collected screen recordings are transferred to the research team's server and automatically searched for all WhatsApp chats. Anonymous log files are created about (1) how long, (2) when and (3) which chat was active. These log files are used for further analysis. The research team does

not view your screen recordings manually but has access to your screen recordings on the server.

Based on the response, we ask more detailed questions related to the reasons for declining participation in every scenario. The respondents can choose several answers. These include reducing the time span of data collection, not collecting data about chats, collecting only non-private data, or no recording of the screen or sending the recordings to a server (see fig. 5). We recruited 35 students from an undergraduate computer science class to participate in our survey. This target group is able to understand the technical differences of the presented methods and has an overview of related privacy issues.

5.5.2. Technical Evaluation

Parallel to the evaluation of our privacy concept we evaluate the technical implementation of our application to enable further long-term studies. Besides an efficient privacy concept that can convince participants, we have to ensure our application can provide stable and reliable log results on a mobile device over an extended period of time under real-world conditions in a pilot study. We collect data on a Google Pixel 2 for two weeks. The questions are: what kind of data can we generate and how does our application perform long-term and in the real-world? We use 30 different event definitions and present general technical results and give an overview of the produced log files.

Additionally, as the permanent recording of the screen and generation of log files from this is a compute-intensive part that can conflict with the stability and energy management of the OS and device, we did test runs on 10 different devices and setups for one week The goal was to check how our approach performs on different devices (as there are a lot of different Android devices and setups) and to find device-specific problems and requirements. Our criteria were energy consumption, the performance of the device while running the application, and stability during the test phase.

5.6. Results

First, we present the results from our online survey comparing different scenarios for collecting usage data on mobile devices. In the second part, we give

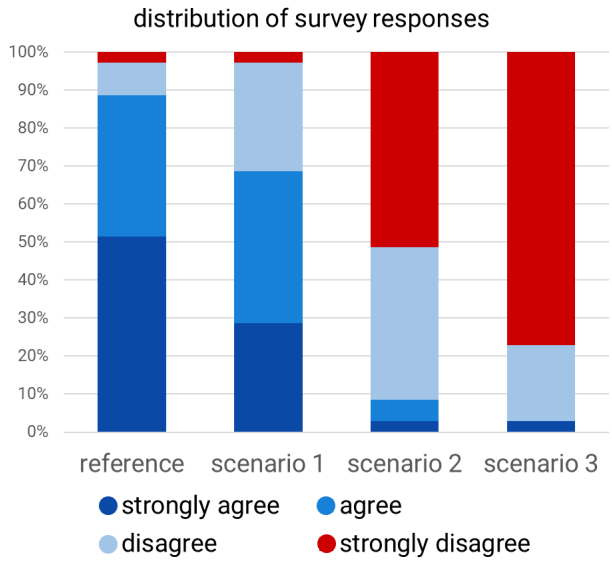

Figure 5.4.: This graph shows the results of our survey, which examines how high the willingness is to participate in a study that uses one of these three described scenarios. The first plot shows the general level of willingness of subjects to participate in scientific studies.

an overview of testing our implementation on ten Android devices for one week to generate log files and outline recommendations when using this approach in user studies.

5.6.1. Survey on Privacy Concept

Our online survey aims to explore how convincing our privacy concept is for potential subjects to participate in a study using our logging approach. We compare our method to two other scenarios: one that is less privacy-invasive and works without screen recordings, and one that is technically comparable to our approach but does not have a privacy concept and transfers the recordings to a central server. We visualize the results in figure 5.4 and list the answers on what could make participation more likely for respondents in figure 5.5.

We have a sample size of 35. The first question after the survey introduction asks for the participants' willingness to participate in scientific studies in general, to filter out respondents who would in general not participate. The median for this initial question equals "strongly agree" (see results for "Reference" in figure 5.4). One participant does not want to participate at all in sci-

entific studies (strongly disagree) and three would rather not (disagree). The one respondent who is negative about participating in studies, in general, is also negative about any further questions and does not choose any of the suggestions offered that would increase willingness to participate.

Most respondents (24) would agree to participate in a study using scenario 1, which does not utilize any screen recording, the median corresponding to "agree" on our scale. For 14 participants a period shorter than a week would increase their willingness to participate (see figure 5.5). Besides, 15 of the survey respondents indicate that they would be more likely to participate if no chat partner data or no personal data were collected.

Eighteen participants (51%) strongly disagree and 13 (40%) disagree on participating in a study utilizing our approach presented in this paper (scenario 2). Thirteen of them state that the type of data collected (chat partners, personal data) is a reason for them not to participate. Twenty-six say that they would be more willing to participate if their screen were not recorded. The time span is specified as a factor by only two of this group. Three respondents would agree and strongly agree on participating in a study using this scenario for data collection. Two of them state a shorter period (3 hours) would increase willingness to participate.

The median result for scenario 3 equals to 'strongly disagree'. Twenty-four (77%) participants strongly disagree and seven (20%) disagree on participating in a scenario 3 study. The most often (27 times) stated answer is that the scenario would be more attractive if the screen recordings were not sent to a server. Following this, 24 respondents state that they would be more likely to consider this study design if no screen recordings were stored on the device. The time span again only played a role in the willingness of 9 of the participants. For 11 participants the study would be more attractive if no personal data or no data about their chat partners were collected.

5.6.2. Technical Evaluation

Two-weeks Pilot Study

The energy, CPU and memory consumption while recording was low, as expected (Krieter & Breiter, 2018a) on a Google Pixel 2. The battery consumption was not noticeable. This should hold true for most current high-end smartphones. The CPU usage while generating the log files from the recorded video material stays around 15% and using around 170 MB of memory. Battery con-

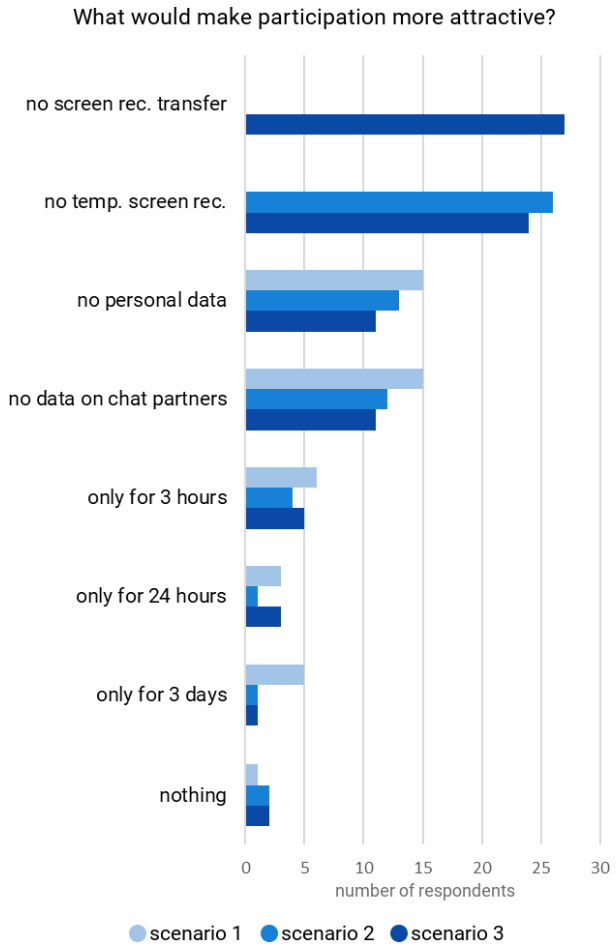

Figure 5.5.: This chart shows the frequency of responses that participants gave to the multiple-choice question of what would make participation more likely for them. Multiple answers were possible.

sumption is not relevant as this task is only performed when the device is plugged in and the screen is off. The consumption of bandwidth is moderate with 156 Megabyte over a period of 14 days to transfer the log files produced. We collected log data of 1077 screen recordings of phone usage which results in 22 hours of screen-on time with a mean screen-on time of 1 minute and 13 seconds per usage. On average we see turning on the screen 76 times and 1.5 hours of use per day. Most of the produced log files were generated and sent to our server overnight. Only in rare cases was the device plugged in and not in use during the day. The applications background service for recording and event detection got killed one time during the data collection due to low battery power. After the data collection phase, there were 349 screen recordings (6 hours 35 minutes) left on the device to be processed (taking another three days).

Resulting Log Files

In total, the log files contained more than 1.5 million entries for all video frames. Consecutive frames with the same log message were summarized as one event with a starting time and a certain length. This resulted in 7774 events of 65 unique log events. We see phone calls with 10 different people, all were called one time, besides one person that was contacted 5 times successfully. One of the most present applications in our log data is WhatsApp with 3004 events. WhatsApp-related events are found in 78% of all usage sessions. Of these events, 1627 times a chat with one of 12 chat partners was opened. The most frequently used chat was opened 1239 times in total, while the second most frequent chat was opened just 85 times. A picture or video was taken 9 times in a chat. The rear camera was in use 30 times. The difference might result from switching from the camera to an album to select an existing picture for sending. We did not define any events for this case. In comparison, the SMS chat was rarely used with only 6 events, chatting with 4 different people. There is an overlap of 2 chat partners with which both Whatsapp messages and SMS were sent. The Instagram feed was opened 369 times, in 13 cases there were unread chat messages. For chatting, Instagram was used 24 times with two chat partners, who are both also present in Whatsapp chats. The BBC News application was used 14 times at 7 different days to view the top stories. Looking at the more general operating system and UI events, we found 498 events for pressing the k-key on the keyboard for example. The "keyboard opened" event was detected 2628 times.

Test Runs on Different Setups

The range of test devices contains eight smartphones and two tablets (manufactured 2014 to 2017), which are in use on a daily basis. The following list specifies the test devices and summarizes the results of the one-week test runs.

Not stable test-runs:

- **Huawei p10 lite,** Android 7, 2017, 4GB RAM
  The app got killed on the first day, the third day and again on day five. The reaction time during the recording was slightly slower, animations were described as a little delayed. Apart from that, the application worked as expected.

- **LG G5,** Android 7, 2016, 4GB
  The application crashed two times in the very beginning but was stable for the rest of the week.

- **Samsung Galaxy A3,** Android 8, 2018, 4GB RAM
  After the first 24 hours, the application crashed for the first time and kept crashing every one or two days for the rest of the test run duration.

- **Benq Aquaris X,** Android 8, 2017, 3GB RAM
  The device storage was full after a couple of days which stopped the application. The UI reaction time during the recording was slightly slower.

Stable test-runs:

- **Samsung Galaxy Tab S3,** Android 8, 2017, 4GB RAM

- **Samsung Galaxy S5,** Android 7, 2014, 2GB RAM

- **Motorola Z Play 1,** Android 8, 2016, 3GB RAM

- **Google Pixel 2,** Android 8, 2017, 4GB RAM

- **Google Pixel 2 XL,** Android 8, 2017, 4GB RAM

- **Samsung Tab A 2016,** Android 7, 2016, 3GB RAM

We see that on 6 out of 10 test devices our application worked as expected for a test run duration of one week. On four devices we recognized that the application got killed or crashed multiple times. The hardware specifications of our test devices do not play an important role as most devices are fast enough to record video in the background. CPU and memory consumption while recording was low as expected, this confirms previous findings by Krieter & Breiter (2018a). The same holds true for battery usage. On two devices, the UI of the OS was slightly slower and less responsive due to the recording of the screen in the background. We classify this as a software side aspect as on other devices that have less powerful CPUs, GPUs and memory the recording is not noticeable and the load for recording stays around 1%(CPU, measured with Android Studio). The energy management of Android and sometimes additional software from the manufacturers are strict about applications that work in the background. Although the application is doing the compute-intensive parts when the device is plugged in, the background service might get killed from time to time. Removing the application from being watched by the OS and energy management helped. On one device the storage was

full which led to the application stopping. Although the screen recordings are deleted from the device after processing them, the application still needs enough space temporarily, to record and save videos for one or two days. This also depends on screen-on time and when the device is plugged in as this is the time window in which the event detection happens. Original Google Android without third-party modifications by manufactures works more stably for our application.

5.7. Discussion

From the point of view of the study participants, it remains very problematic when the screen of one's own private device is filmed. Using screen recordings requires a high level of trust and centralization is not the preferred model in terms of privacy (Tang et al., 2006; Spiekermann & Cranor, 2009; Barkuus & Dey, 2003; Hong et al., 2003). The data is both large and highly sensitive. Therefore, processing the video data on the device without transferring it over the Internet is desirable. By moving the detection of events from the server to the client side, the users or participants do not have to share any screen recordings with the research team. A current trend is to process sensitive data locally: Google for example plans to save and process all face recognition related data and steps on its Pixel 4 locally (Barbello, 2019).

The intention of our privacy concept is to make our approach privacy-friendly and make it easier to find participants for HCI research using screen recordings. The evaluation of our privacy concept in a survey study gives us hints that our implementation increases privacy for participants, and slightly the willingness to participate in research studies, compared to using a server-side implementation. But recording the screen permanently still seems to be highly problematic for most respondents of our survey, even if the video data is not evaluated manually by the researchers, or transferred to a server. By this, the results follow insights and experiences from previous research involving screen recording and private data over longer time spans(Reeves et al., 2019; Tang et al., 2006). Only three participants say they would agree on participating in a study using the logging application presented in this paper. The most stated reason by the group that declined to participate is that the screen is being recorded. Even though the results here are slightly better than with the server-side version (which only one participant would

accept), the comparison with scenario 1 without screen recording shows that this scenario is much more likely to be accepted by our respondents. Our research scenario for the survey is very privacy-invasive (on purpose) which might result in respondents to declining potential participation. Thirteen participants state that a study using our approach would be more attractive if no personal data or data about their chat partners were collected. Messaging is especially considered problematic (Reeves et al., 2019). Following this, the approach might be usable for research scenarios that involve only little or no personal or private data. By processing the videos on the device and only sending anonymized or pseudonymized log data to the research team's server, the privacy situation improves. It still has to be transparent for participants, which events the application is searching: researchers need to address privacy concerns carefully. When defining events, they must take into account if these events might reveal private or personal data or identify the user.

The results from the pilot study show that the approach can generate log data under real-world conditions over a period of one week on a range of devices. The energy and resource consumption during recording is acceptable on the test device. By running the event detection on the mobile device, it is easier to scale our approach to studies with many participants, compared to server-side processing. Transferring screen recordings over the internet to a central server for analyzation uses a lot of bandwidth and results in a high computational cost for the server. The event detection only happens when the phone is plugged in and not in use, which limits the time for processing the recordings. The 'in time' generation of log files depends on how many events we look for, how much time we record per day, how much idle time for processing we have, and the efficiency of the implementation and the device itself. Another solution might be to combine the presented approach with more general logging on mobile devices that rather track general system events (e.g. Ferreira et al., 2015). With that, we could check which application is open using system events and then search for specific in-application events using screen recordings.

By doing the event detection on the mobile device, we have to know what we are looking for upfront. The events we are looking for have to be defined before we collect data. When collecting screen recordings first and transferring them to a server, we can use the video material again and again and adjust our event definitions to gain better data. For research projects it might be a good approach to first collect some pilot screen recordings, analyze them

manually, define events and validate the results. After that, we can roll out the privacy-friendly variant in the user study.

5.7.1. Limitations

Although the evaluation results on our privacy concept from the survey are consistent with previous work and literature, a larger sample size would help strengthen the reliability of the evaluation.

The presented approach is based on the visual screen output. This expands our logging capabilities to generate data in many situations where we could hardly gather data before. However, it also limits the log file generation to visible interactions.

Another limitation is that our approach only runs on Android (although the market share of the OS is over 80% worldwide (IDC & Gartner, 2018)). For generating log files in the same manner on iOS, for example, another implementation is necessary. We identified problems with four of our ten test devices. We need to further improve the stability of the approach on different devices and OS combinations to make the approach usable in future research. For further improving scalability, the implementation can be improved by utilizing multiprocessing on compatible devices for example.

5.8. Conclusion and Future Work

This paper contributes to expanding usage possibilities of the rich but privacy-invasive data source screen recordings for HCI research. The idea of this research is, on the one hand, to develop a privacy concept that can convince potential participants to take part in research studies involving screen recording and on the other hand to achieve the technical prerequisites to scale the use of this method easily. The following list sums up the key contributions of this paper.

- We developed and implemented a privacy concept that makes the use of screen recordings as a long-term data source more feasible in real-world studies, mainly by shifting the detection of events to the mobile device and anonymizing parts of the generated log files. The evaluation in a user survey indicates that our approach is more likely to be accepted

in a user study, compared to a server-based version, but still, potential participants are hard to convince to be part of studies that involve long-term screen recording.

- In a two-week pilot study we show that our implementation can provide long-term (for screen recordings) results under real-world conditions and give an example of what kind of data we can generate. Additionally, a one-week test on ten different mobile devices setups indicates that we can run the current approach on several Android devices. We discuss issues of our current implementation in terms of performance and stability.

For future work, we will use the results of this paper to conduct long-term user studies on mobile human-computer interaction to generate data in situations that cannot be covered by other logging methods. Further development might include using the approach for the 'live' detection of events. By using only one frame per second, we could directly detect events and interact upon these with the user. A possible scenario would be to ask a question about a specific action or an intervention, for example.

The source code of this project is available open-source to the community (Krieter, 2018c).

This work was funded by the German Ministry of Education and Research (reference number: 01JKD1709B).

Part II.

Application of the Approach in the Context of Education

6. Log files for Learning Analytics

This chapter has been published under the title "Track every move of your students: log files for Learning Analytics from mobile screen recordings" in the proceedings of the 16th E-Learning Conference in Computer Science (DeLFI'18) by the German Informatics Society e.V..

6.1. Introduction

The field of Learning Analytics uses a wide range of data sources. A common data source are log files from systems that support the learning process like Learning Management systems (Papamitsiou & Economides, 2014). Pardo & Kloos (2011) state that early Learning Analytics research is "LMS-centric": using only data from an LMS might limit research to a small part of the activities of students. Especially communication often happens through existing email or mobile chat applications and not through the communication features of the used LMS (Pardo & Kloos, 2011). This reduces the informative value of log files from LMSs. To provide a comprehensive picture and analysis of learning activities, additional data is helpful beyond the log files of the LMS. In order to analyze learning outside the LMS, additional data is necessary, especially when various third-party applications and software are used in a digital learning environment. In these applications, it is difficult to gather data about the interactions of students. To get log data from these third-party applications, it is usually necessary to implement log commands in their source code. However, many applications are not open source and changing the source code to generate data for Learning Analytics is not possible. Papamitsiou and Economides state that "[..]every 'click' within an electronic learning environment may be valuable actual information that can be tracked and analyzed. Every simple or more complex action within such environments can be isolated, identified and classified through computational methods into meaningful patterns." (Papamitsiou & Economides, 2014). Hepp et al. describe

these small pieces of data we leave behind when we act in digital environments "digital traces" and stress that these traces are meaningless until we put them into a relevant context by appropriate methods and their triangulation (Hepp et al., 2018). While we leave more traces in digital learning environments than ever before, it is still a challenge to collect these data points in an appropriate way and give meaning to this data through adequate Learning Analytics, for example. Our work addresses this lack of data sources in digital learning environments containing not just an LMS but also multiple closed source code applications by presenting a method to generate log files for Learning Analytics based on mobile screen recordings. A graphical user interface (GUI) always represents the current state of a system, based on user and system events (Aho, Kanstrén, et al., 2014). The focus on GUIs in Human Computer Interaction (HCI) leads to the point that the analysis of screen recordings makes it possible to understand almost every interaction, every behavior or every task that the user performs while using a computer system. Interpreting screen recordings manually is very time consuming, what limits this approach to relatively short time periods. Our approach utilizes computer vision and machine-learning methods to automatically analyze screen recordings and to create event-based log files. Our intention is to combine the advantages of short-term detailed screen recording analysis with long-term log file generation. The main contribution we aim at in this paper is to introduce a new method to collect data in digital learning environments. We present how we define events of interest and find these in mobile screen recordings in order to create log files for further analysis. The results and evaluation in this paper are based on 118 example screen recordings from a mobile device. Besides that, we show basic descriptive visualizations of example data and show how this could be used as a new data source for Learning Analytics.

6.2. Related Work

6.2.1. Data Sources for Learning Analytics outside the LMS

A systematic literature search by Papamitsiou & Economides (2014) identifies various data sources used in the field of Learning Analytics and educational data mining, such as log files of goal-oriented implemented systems, questionnaires, interviews, web tracking software, open data sets and vir-

tual machines. A main source for data in the field of Learning Analytics are still log files containing entries about interactions of students within an LMS (Tempelaar et al., 2015; Pardo & Kloos, 2011). There are several efforts in the Learning Analytics community to collect data on student behavior and interactions outside the LMS. Pardo & Kloos (2011) present an approach using virtual machines to log a number of events outside of an LMS. They provide pre-configured virtual machines to students of a programming class that the students run on their personal computers. The machines monitor a couple of events like power-up and shutdown, start and use of selected tools and commands and browser history. They show that a significant amount of relevant interactions happen outside of the sphere of the used LMS. Another attempt to gather data beyond the LMS is introduced by Kitto et al. (2015) utilizing data from social media combined with LMS data. They present an open source toolkit accessing six social media APIs and save in a standardized actor-verb-object notation for further analysis. Their focus is to gather data from multiple sources in a uniform way at large-scale, but also concerns regarding related privacy and data ownership issues. Another study by Tempelaar et al. explores and compares the predictive power of different Learning Analytics data sources (Tempelaar et al., 2015). Therefore, they collect self-reported data, LMS data and e-tutorial data of formative assessments. In the conducted study, data from 922 mathematics and statistics students is collected. They mention that the use of only LMS data does not have substantial predictive power in their study. They find that the use of formative computer-assisted assessments is a good predictor for detecting underperforming students. There is still no agreement within the Learning Analytics community as to which interactions of students within a digital learning environment are decisive for effective learning (Agudo-Peregrina et al., 2014). More research and exploration of different data sources is needed, to address this. Coming from this point, our approach, which we present in this paper, follows the path of opening up and developing new data sources with the aim of providing a more comprehensive picture and a new perspective on learning activities.

6.2.2. Detecting events in screen recordings

The analysis of screen recordings of student devices is not yet used in the context of Learning Analytics, neither for manual nor for automatic analysis. There are some research approaches that focus on the automatic analy-

sis of screen captures, but not in association with the generation of data for Learning Analytics. Most research in this area is aiming at analyzing Human Computer interaction, but there are also approaches that could be used as a data source in other contexts such as learning analysis. The project "InspectorWidget" (Frisson et al., 2016) proposes an automatic screencast annotating system for usability checks using computer vision and machine learning techniques. The approach is designed for usability checks and requirement engineering scenarios. The work is in an early stage, but has a potential for long-term research studies and generating log files for several purposes on desktop devices. The project has been developed for Linux, Windows, and Mac, but lacks to support mobile devices like smartphones and tablet computers. An approach by Chang et al. (2010) has some close relation to our approach although their aim is not to generate log files from screen recordings. The focus of their project "Sikuli Test" is to test desktop GUIs automatically using computer vision methods. The script acts "like a robotic tester" and acts on the visual screen output of desktop GUIs based on previously defined events. Common with our approach is to detect individually defined GUI states using computer vision methods. Matejka et al. (2013) introduce a tool for the collection of data about software usage, with effort spent in making the approach independent of the actual application. This aim is closely related to our approach, but the way of data collection is different. The result of this is a dynamic heat map overlay, that shows usage patterns of the active user and of the community. They collect data about how frequently users use functions in office applications aiming at optimizing software. The tool is only available for Mircosoft Windows, using its accessibility APIs (not available for all applications), making the method useable for many Windows applications, but requires additional work for every application (Matejka et al., 2013). Based on the previous work on data sources in Learning analytics and we present how we combine computer vision and machine-learning methods to automatically detect events in mobile screen recordings to generate log files for Learning Analytics.

6.3. Technical Approach

The process for generating log data and further analysis in form of Learning Analytics involves several steps (see Fig. 1). At first, video data needs to be col-

lected on the student's devices, followed by the definition of events of interest and automatic analysis of the video material based on these event definitions. The resulting log files need to be prepared for further analysis, to be finally analyzed and put into context.

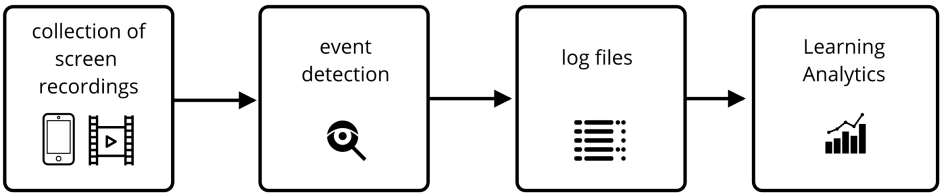

Figure 6.1.: Simplified process: from screen recordings to log files for Learning Analytics.

6.3.1. Collection of screen recordings

To validate the results of our approach we recorded 118 smartphone usage sessions (68 minutes), which is roughly equivalent to the average use of a smartphone during one day (Böhmer et al., 2011). With this amount of video, it is still possible to manually validate the log results and identify errors in the detection process. We define a phone usage session as from switching on the screen until the screen turns off again and record interactions in the locked state as well as in unlocked device state. For evaluation of our approach, we recorded high-quality video in full HD at an average bitrate of 4418 Kbit/s and 20 frames per second (FPS) on an Android device.

6.3.2. Event detection

Since video analysis is a compute-intensive process, optimizations that reduce the workload are of importance. As our focus are screen recordings from mobile devices, we can exploit some characteristics of the mobile platform. In contrast to most desktop systems, mobile devices have a way more structured and fixed GUI. On desktop platforms, it is common to have multiple resizable windows and applications opened at various positions on the screen. Mobile platforms usually only support to display one application at a time which lowers the effort to check where and which application or activity is present. Be-

sides that, most applications follow the design guidelines of iOS and Android, which gives GUIs a fixed structure and makes them more predictable.

Video Preprocessing

To speed up the detection process we slightly preprocess the video material, as there are many frames without a visible difference. At a frame rate of 20 FPS, we skip 54

Definition of Log Events

Events of interest are defined on the basis of GUI states and contain a message that is written to the log entry. For our test runs, we defined a list of 30 events ranging from general user interface (UI) events like "keyboard opened" to more specific events like "Whatsapp chat with Person A". Figure 2 shows an example of an event definition of a Whatsapp chat. Every event can consist of different elements: comparison of fixed image areas, search for image parts in the whole frame and comparison of text elements or reading text from the frame. An event definition can contain multiple fixed areas. These image parts are compared to the same area of every video frame. This exploits the fixed structure of the mobile platform, as it is not necessary to search in various positions on the screen in many cases. The similarity is determined by a perceptual hash function (Buchner, 2018). Perceptual hashes are rather robust and work with different image scaling or artifacts caused by video compression. To search for an image part that can appear at any or not a fixed position of the screen we use OpenCV's (OpenCV library, n.d.) template matching algorithm. As this step can cost more time in contrast to comparing fixed areas it is more efficient to use fixed areas if possible to define an event.

We are searching for text strings or reading text from a certain area of the frame using optical character recognition (OCR). For this, we use the open source library tesseract (tesseract-ocr, n.d.). Text recognition is helpful to distinguish a variety of events. An example for fetching text from a frame is the name of a chat partner that is attached to the log message (Fig. 2).

Detecting Events

The detection process works frame by frame and in a certain order, that keeps the computational effort low. Every frame is checked against the list of events. A frame can contain multiple events or no events. The first step is to compare all fixed areas from the event list against the frame, as this step is fast, compared to template matching and OCR. After that, template matching is

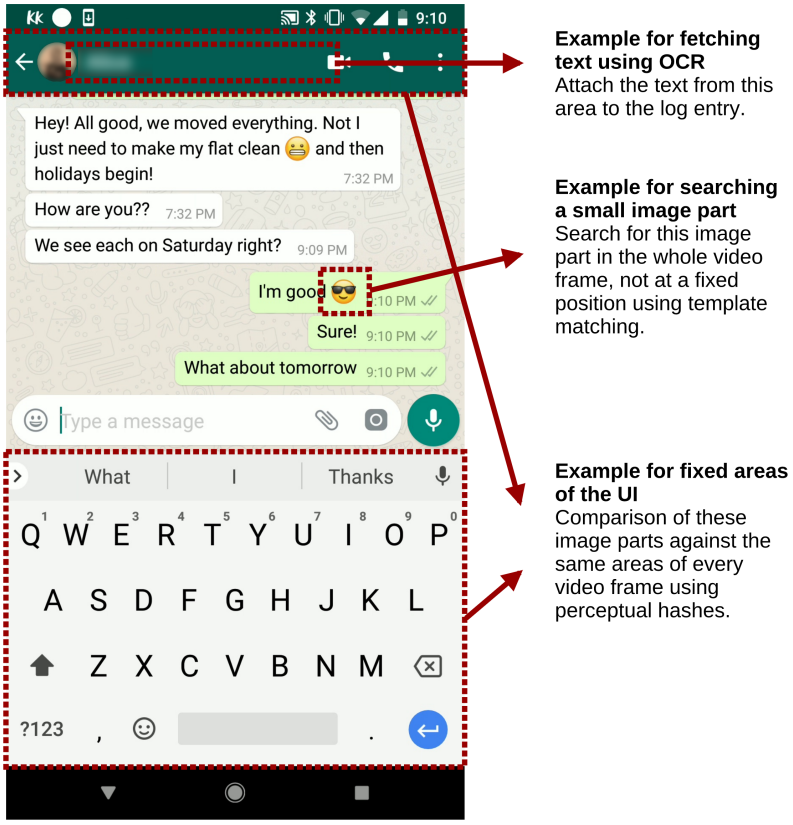

Figure 6.2.: Definition of an Event. In this example a text string is fetched from an area at the top, two fixed areas are set, one at the bottom and one at the top (keyboard and chat header) and we search for a non-fixed image part in the whole frame (smiley).

executed followed by OCR as the last step. As each frame is processed individually, it is possible to scale up the detection process on multiple CPU cores.

6.3.3. Log files

The resulting log files contain one or more entries for every video frame. An entry consist of frame number, timestamp and log message. To further work with these entries we transfer these logs into a different formatting, to make further analysis more feasible. The final log files are event based instead of frame based. E.g. if 85 consecutive frames contain the event “Instagram feed opened”, this would lead to one event “Instagram feed opened” with a timestamp and a certain length. Listing 1 shows example log entries from the resulting final log file. The last log entry for example indicates that a Whatsapp

chat to a certain person was opened for 31,4 seconds at 17:37:14 on the 14th of December 2017.

```
1513269377027.0;2017-12-14 17:36:17.027000;2000.0;
    whatsapp, chatlist active, number of chats with new
    messages - 1
1513269377227.0;2017-12-14 17:36:17.227000;1800.0;
    whatsapp, chatlist opened
1513269379027.0;2017-12-14 17:36:19.027000;800.0;whatsapp
     chatlist, scrolled down
1513269380027.0;2017-12-14 17:36:20.027000;3400.0;
    whatsapp, chatlist active, number of chats with new
    messages - 1
1513269380027.0;2017-12-14 17:36:20.027000;3400.0;
    whatsapp, chatlist opened
1513269383627.0;2017-12-14 17:36:23.627000;12200.0;
    whatsapp chat opened - Luca
1513269386027.0;2017-12-14 17:36:26.027000;6200.0;
    keyboard opened
1513269394027.0;2017-12-14 17:36:34.027000;2000.0;
    keyboard opened
1513269395627.0;2017-12-14 17:36:35.627000;200.0;
    homebutton pressed
1513269396027.0;2017-12-14 17:36:36.027000;2400.0;home
    screen present
1513269398427.0;2017-12-14 17:36:38.427000;32200.0;
    Instagram opened
1513269398427.0;2017-12-14 17:36:38.427000;32200.0;
    Instagram, feed opened
1513269398427.0;2017-12-14 17:36:38.427000;32200.0;
    Instagram, feed opened, unread messages
1513269434427.0;2017-12-14 17:37:14.427000;31400.0;
    whatsapp chat opened - Luca
```

Listing 6.1: Example entries from the result log file. All entries contain a timestamp, a length in milliseconds and a log message that describes the event.

6.4. Results

Altogether 79790 frames were processed from the 68 minutes of video material and a log file with one or more entries per frame was generated. This video frame based log file was transferred into a different formatting as described above (see listing 1). In total, the final log file consisted of 931 correctly detected events. The result log file did not miss any events but contained 1.3 percent false positive events using the originally collected high-quality video material. We define a false positive as an event with a correct log message

that did not occur in the video material. The most problematic event definition was the event “calculator opened”. The calculator was only opened one time in all videos but was detected falsely 11 times. The reason for these false results was an imprecise definition of the calculator event. Another reason for several messed up log messages were events that fetched text from transition frames using OCR. On these frames between switching or closing applications it is not clear which application is present, as both are morphed into each other. These animation frames caused messed up names of chat partners when the chat applications were closed, for example. Figure 3 illustrates a usage session containing several application usage sessions. We visualize the result log files in reference to van Berkel et al. (2016) as usage sessions containing detailed application usage sessions. As an application session, we take the time from opening and using an application until the application is closed again or switched. The upper timeline shows application events, the lower blue line shows the use of general UI events, which can occur in parallel to in-app events. In this example, the events “keyboard opened” and “k key pressed” occurred several times.

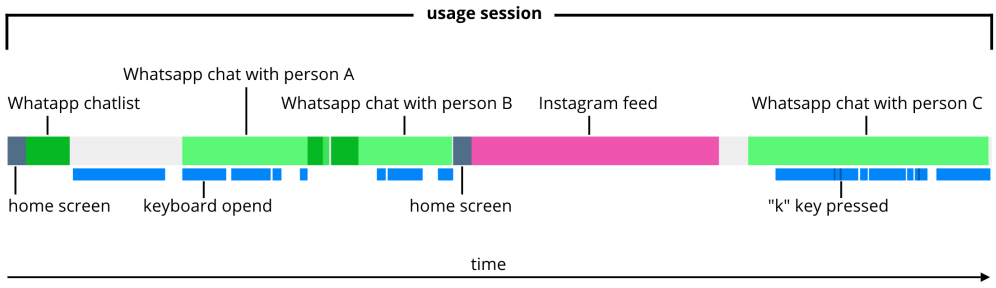

Figure 6.3.: Visualization of a phone usage session containing the use of different applications and in-app interactions.

This usage session shows the use of two different applications (Whatsapp and Instagram) and contains several detailed in-app events about the use of Whatsapp. Three chats with different partners were opened, partly together with the presence of the keyboard. At least in the last chat, the keyboard was opened and used. Most likely in the other cases as well, but for testing we only defined an event for pressing the k key. Most test events were connected to chat communication. In addition several events regarded the general UI, news applications, and Android system events such as lock screen interactions or displaying the ”all applications” menu for example. The result log file contained detailed information about chat communication in four different ap-

plications with a chat function: Whatsapp, SMS chats, Instagram and Pinterest. For all chat application events the name of the chat partners was saved. In total there were chats with 13 different persons, the most conversation happened through Whatsapp. One chat partner occurred in all four applications. The event definitions for Whatsapp were most detailed, ranging from taking a picture in a chat (happened two times) to scrolling down the list of chats and number of new messages. Besides those two phone calls were registered to the same person, one missed call and one outgoing call. Other events included using the phone's contact list, creating a new contact, viewing the "top stories" in the BBC news application, using the calculator, and whether the device is used in unlocked or locked screen mode. All event definitions besides one used at least one fixed area and perceptual hash comparisons. Template matching was defined in two events and OCR in 12 events. In general comparing fixed areas worked reliable for our purpose, as well as template matching. The OCR results were very accurate, unless used on transition frames during application switching or closing. We used multiprocessing on a dual-core Intel i7 for all test runs. Processing all 118 high-quality videos took about 10 hours. Ideas for speeding up the process are briefly outlined in the discussion.

6.5. Discussion

Most data sources for Learning Analytics rely on log files from LMSs or related systems. These data sources have two clear limitations: log events cannot be individually defined in most cases, unless the source code is modified and most the log files are limited to interactions within the system. We address this challenge by introducing a logging approach that relies on the visual screen output. Theoretically, it is possible to track everything that is visible on the student's device screen, leading to data about interactions in very high detail. Log events can be detected in multiple applications while only using one method to generate the log file which eliminates the need to merge log files from multiple sources. Instead of implementing log commands at development stage, researchers can define events of interest based on GUI states instead. This can be done before collecting video data or even after. In case that the logged events are not useful or predictive for the research question, event definitions can be adjusted and the video material can be analyzed again from a different perspective. Another possibility resulting from the use of this ap-

proach is that no access to the source code of the applications is required to implement log commands. This opens a new way to log in third party applications, since many applications are closed and do not produce any log files.

6.5.1. Privacy and Ethical Challenges

A serious challenge before applying our approach in a real-world scenario is possible, is the privacy of student's personal data. The student's device screen is permanently recorded, saved, and analyzed. This requires an informed consent by participants and raising the question of how participants in research studies can be informed what data they share for what purpose and how they can control what they share. We follow the Privacy by Design (Cavoukian, 2011) approach in order to develop an effective concept for the protection of the privacy of participants already during development. A step in this direction would be that the screen recordings are not be accessible to researchers and would have to be approved by users to be searched for events. Additionally, the anonymization of the resulting log files is obligatory, before they could be used in a Learning Analytics context. With regard to these serious obstacles to privacy, ethical questions also arise as to whether the gathering and use of these log files justifies such a data collection and analysis. This ethical perspective will depend greatly on the success and implementation of a sustainable privacy concept, that ensures the privacy rights of students or participants and is in accordance to current regulations.

6.5.2. Technical Challenges

For further analysis of learning paths, student behavior or predictions based on these log files, it is essential, that the data quality and correctness is ensured. The result logs of our sample video material reached a high level of correctness for 30 test events, ranging from very short events like pressing a key on the keyboard to reading in a news app. We identified some event definitions that produce unexpected or messed up results and incorrect log messages that lead to false positives, though no events that occurred in the test video material were missed. Another technical challenge besides correct log results is the great file size of the video material and computational effort to process the screen recordings. The 118 sample screen recordings we used for

this paper to validate the results of our approach as a data source added up to 2,2 GB. Since the recognition works frame by frame, the recognition process can be considerably accelerated by multiprocessing. Besides scaling the number of CPU cores, it seems reasonable to reduce the video frame rate already at recording time, since 54% of all frames were duplicates or almost duplicates.

6.6. Conclusions and Future Work

In this paper, we presented an approach for a new data source in Learning Analytics that works beyond an LMS, in multiple applications and independent of the applications source code. We showed that we could generate detailed log files based on mobile screen recordings and track a range of interactions on mobile devices. We demonstrated examples of how this log data can be further processed to serve as a basis for Learning Analytics scenarios from a new perspective. The next steps involve the development of privacy measures, performance and result optimizations. It is also necessary to investigate how this data source can be used sensibly, what insights are possible, and how this approach can be reasonable from a viewpoint of research ethics.

7. Estimating Students' Online Time by Combining LMS Log Files and Mobile Screen Recordings

This chapter has been submitted as a journal article with the title "Are You still there? Estimating Students' LMS Online-time by combining Log Files and Screen Recordings" to the IEEE Transactions on Learning Technologies.

7.1. Introduction

Log files of Learning Management Systems (LMS) are a frequently used data source in the field of Learning Analytics (LA) (Papamitsiou & Economides, 2014). Several measurements of student activities are derived from this data, like the number of clicks, logins and time spent using the LMS (e.g. You, 2016; Conijn et al., 2017; Kovanović et al., 2015). These data points form the features for further statistical analyses or predictions about the success of a student when attending a course and the potential need for support for passing the course, for example.

This makes the process of how we derive and define these measurements from the raw log data highly important, as further decision models are based on these data points. One of the most important and frequently used dimensions is the duration that students spend online in the LMS (Conijn et al., 2017; Kovanović et al., 2015; Dawson et al., 2008; Damianov et al., 2009). LMS log files usually do not directly define or track the duration of the time students spend in the system (Kovanović et al., 2015). For this reason, it is common to use heuristics to summarize a series of clicks in a session and calculate the usage time, e.g. based on the login and logout clicks (Sael et al., 2013).

Usually, web log files are based on clicks (HTTP requests) and therefore stateless. It is not possible to clearly say if a user is really still using the LMS, or maybe left the browser or tab to do something else and continue using the

LMS later on. This means that if we calculate the usage duration of a session based on the log files, we can not say for sure if the LMS has been used for the entire time that we calculated based on a series of user clicks. It is difficult to define this construct because the data does not say directly what we want to know. We cannot keep track of off-task behavior from the LMS logs. This results in calculating a session duration that does not reflect the actual online time (Sael et al., 2013). In data pre-processing for LA there are several definitions to calculate the session duration in the literature (Zacharis, 2015; Conijn et al., 2017; Sael et al., 2013; Ba-Omar et al., 2007; Kovanović et al., 2015). However, it is essential to define and calculate this measure as accurately as possible, since session duration is a frequently used feature in learning outcome prediction models. Kovanovic et al. show that different strategies of estimating the time spent with the LMS can have a great effect on the produced statistical model and predictions (Kovanović et al., 2015).

We address this challenge by combining two data sources to conduct an empirical investigation of the duration of LMS sessions. We collect log files from the LMS "Moodle" and record the screen of the students' tablets in parallel in the background. We then use existing session duration definitions from the literature to calculate the session duration of our students' Moodle logs and compare the results with the actual duration that the LMS was present in the students' screen recordings. We collected data for four months, resulting in a data set of more than 19,000 Moodle log entries and over 10,000 minutes of screen recordings. Our approach is based on computer vision and machine learning of Krieter and Breiter (Krieter & Breiter, 2018a,b) to automatically generate log files from our screen recordings and extract the Moodle usage time from the large amount of video data. In addition, we show that depending on the calculation of our LMS session duration, the number of sessions also changes considerably.

Our main contribution aims to investigate and enhance the pre-processing of LMS log data for LA. We present a methodological approach to estimate the online time duration based on two different data sources and discuss the implications and limitations of our methodology. Using counter-examples, we show that common definitions of online time estimation do not lead to precise results. Our goal is not to present a "perfect" estimation definition, but to make the community sensitive towards these assumptions.

- By visualizing and comparing the results from the Moodle log files and from the screen recordings we can show that the duration calculated,

based on the Moodle log file sessions, differs significantly from the time users actually viewed the LMS on their screens.

- Based on this we try to find a session duration definition that provides more exact results for our data set. We use different variations of common session definitions and test them against the video material to find an individual online time estimation definition for each of our students that best reflects their time spent in the LMS. From this perspective, we also take a look at how the number of sessions changes, taking the screen recordings of a user into account when we calculate the number of sessions based on our Moodle log files.

7.2. Background and Related Work

In the first part of this section, we give an overview of research in LA that uses data sources that go beyond LMS log files or combine multiple sources of student data.

Previous research in the LA community has picked up on the struggle over how to define the online duration or time-on-task estimation in a way that works reliably and reflects the actual online time of students. Besides work that specifically focuses on this important step in the data preparation process, we give some examples of common definitions for calculating the online time of students. Table 7.1 gives an overview of several session definitions from the literature that focus on different LMS log files. Most definitions refer to Moodle log files since we also use this LMS.

7.2.1. Data Sources in Learning Analytics

Utilizing LMS log files makes it easy to follow the activities of students in the LMS unobtrusively with low effort. However, relying on LMS data as the only data source also limits the scope of analysis. Several research projects focus on utilizing additional data sources in digital learning environments (Papamitsiou & Economides, 2014), though the most important data source for LA remains log files from LMSs (Tempelaar et al., 2015; Pardo & Kloos, 2011). Which student interactions in a digital learning environment are decisive for effective learning is not decided in general within the Learning Analytics Commu-

nity (Agudo-Peregrina et al., 2014). There are example research projects on utilizing and testing different data sources outside the LMS to gain additional insights on students' activities, varying from data from programming IDEs (Blikstein, 2011), screen recordings (Krieter & Breiter, 2018b), and questionnaires, interviews, web tracking software, open data sets and virtual machines (Pardo & Kloos, 2011).

There are several examples of using data sources outside an LMS for LA in the context of learning programming. Blikstein is using a dataset of a three-week student assignment on programming using a programming environment that logs many users' interactions, such as keystrokes, clicks, variables changes, and changes in the source code (Blikstein, 2011). He shows, how this data can be used to find certain events in the process and suggests identifying situations in which students might need help.

Fernandes-Medina et al. use compile messages as a data source and analyze the work of students to report on the individual and comparative progress of learning (Fernandez-Medina et al., 2013). They use the results to inform students about their learning process. Another recent related approach to this is pursued by Öztürk et al. by developing a web-based programming environment for novice students to collect data (Öztürk et al., 2018). By identifying metrics for student performance they use this data to predict students at risk to drop out at an early course stage. Using screen recordings or screenshots as a data source for LA has been a subject in previous research (Krieter & Breiter, 2018b): the authors present a tool for LA that can generate log files from mobile screen recordings using computer vision and machine learning methods for optical character recognition (OCR) to find events based on the visual screen output.

7.2.2. Online-time estimation in Learning Analytics

Kovanovic et al. present a study focusing on the "black box of time-on-task estimation" (Kovanović et al., 2015). They stress the problem that the time students spend on a task or in the LMS is a commonly used measure, but at the same time is not described in detail and often not accurate. To address this issue they studied the effects of different time-on-task definitions on the results of a common prediction model. They show that the results of the model change significantly, depending on the different time estimation methods. They encourage further research and discussion on this issue. A

study by Munk and Drlik points in the same direction (Munk & Drlík, 2011). They focus on the data pre-processing of log files in education and the difficulties when specifying a time window to define sessions in user logs and the calculation of the session duration.

There are several examples of different definitions of session duration estimation in previous research. Zacharis investigated how students at risk in blended learning courses can be predicted early by analyzing Moodle log data (Zacharis, 2015). For his model, he explored the predictive significance of 29 LMS usage variables. He defined the duration variable as a session of all clicks after the login of a student until logout. In the case of the user not actively logging out to close the session, he ends the session if 40 minutes of inactivity occurs.

A similar method is used by Conijn et al. to define the estimated time students spend online in the LMS Moodle (Conijn et al., 2017). They use the same 40-minute threshold of inactivity to end a session. A session had to consist of at least two clicks, the duration of which is calculated based on the time between the first and the last click. They state that raw log data does not provide concrete measurements and more insights are needed to explore how LMS data can be represented, as well as adding other data sources to add context to the log files.

Sael et al. conducted a study on data pre-processing and using web usage mining methods on a dataset of Moodle log files (Sael et al., 2013). A session consists of all clicks between a user's login and logout in their analysis. They indicate that domain-specific steps in the pre-processing of data for LA are still not explored sufficiently. They reflect on their method of estimating the duration students use the LMS and recognized that there are inconsistencies between the time spent online and the number of sessions done. From that, they derive that students are not following the LMS contents continuously, but switch to other activities while using the system.

We follow the suggestion of Kovanovic et al. to further investigate the methods to estimate the online time of students, which are used to process log data. To gain a deeper understanding of students' LMS sessions we add another data source (similar to Conijn et al. suggestion) to augment our LMS log files and put them into a different context. From the several session and online time definitions, we see that the decisive factor is the time-out variable that closes an opened session. Following this, we evaluate different time-out values of inactivity after the last click of a user (described in detail in the next section).

Table 7.1.: This table lists various definitions for LMS sessions and estimation of online time from the literature.

LMS	Description/Definition	Reference
Moodle	*"[...] sequence of behavior from the first click after the login to Moodle until the last click before logging out, or the last click before staying inactive for at least 40 minutes. Each session consisted of at least two clicks. The time between the first and the last click of a session was used to compute the total time online."*	Conijn et al. (2017)
Moodle	*"For the purposes of this study, an online session was defined as the sequence of a user's interactions with course web page, from entering after login until logging out or staying 40 min inactive."*	Zacharis (2015)
discussion forum	60 minutes time-out and based on the duration of the last action	Wise et al. (2013)
Moodle	*"Session: All user interactions achieved between a user login and a logout."*	Sael et al. (2013)
Moodle	*"In this paper, we used reactive time-oriented heuristic method to define the users' sessions. From our point of view sessions were identified as delimited series of clicks realized in the defined time period. [...] we adopted a 15-minute timeout to start a new session [...]"*	Munk & Drlík (2011)
LTTS	30 minutes time-out, addition of mean duration for last activity	del Valle & Duffy (2009)
custom prototype	*"Session Identification: This task consists of grouping a learner's page accesses in a unit named session by dividing the click stream of each learner into sessions. We have adopted a 30-minute timeout to start new session."*	Ba-Omar et al. (2007)
custom prototype	*"[...] session concept is traditionally understood as the set of user's accesses performed during a site visit"*, no time-out is specified	Marquardt et al. (2004)

7.3. Methods

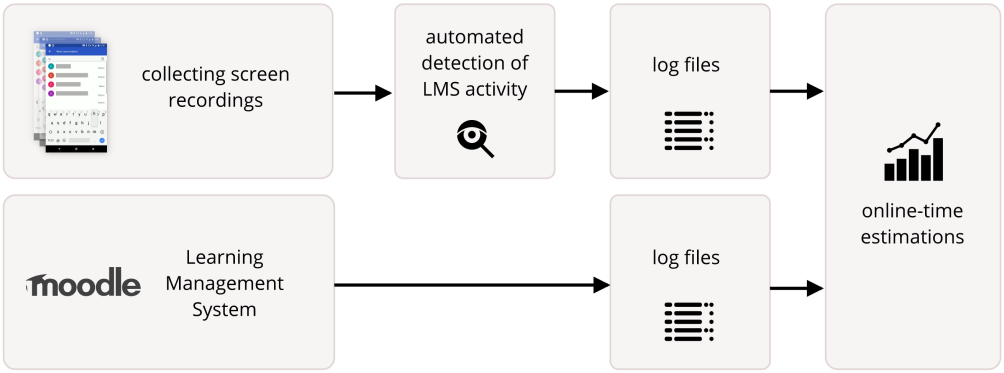

Figure 7.1.: This diagram summarizes our research design and explains how we connect two inherently very different data sources. The goal is to create precise online time estimation for our students.

Figure 7.1 gives an overview of our research design and data sources: we gather data from an LMS and screen recordings. To make both data sources comparable in a quantitative way, we generate log files from the screen recordings and use the log records from both sources to estimate the online time for our users on an individual basis.

7.3.1. Data collection

We collected our data in the context of two blended learning music classes in German Adult Learning Centers (ALC). The duration of the data collection was four months. The topic of these classes was learning to produce music on tablet devices. We used Moodle as a platform to support learning, as well as several applications for music creation. The set-up of the classes was a mixture of formal and informal learning. The teacher and students had a weekly meeting with a combination of teaching and presenting their work and asking questions. Besides that, the students used their devices outside of the class to learn and solve tasks on their own.

Participants

Our course offer attracted nine participants at the ALCs, four females and five males. The ages ranged between 17 and 74 years. The maximum number of participants per course was eight. The recruitment was supported by

the learning centers and ads in local newspapers. All participants received a Tablet (Samsung Galaxy Tab S3), a case with keyboard combination, mouse, and headphones for the duration of the course. As our intention is to show counter examples for online-time estimation the sample number is sufficient.

Privacy and Legal aspects

In order to apply our research design and methods of data collection, we precisely informed interested potential participants about the data collection and analyzing process in our research project. This was necessary to ensure that participants were able to understand all aspects of our research design, enabling them to make a reasonable, voluntary, and informed decision about their consent to be part of our study. Be-sides the necessary requirements by law (based on the GDPR), like informed consent of all participants, for example, our focus was to present our methods and research goals as accessible as possible. We did not expect potential participants to be experts in log file analysis or even familiar with technology at all. Additionally, recording the screen of a user all the time is a very invasive method of data collection that can make it hard to find participants for collecting field data in this way (Tang et al., 2006; Reeves et al., 2019). Tang et al. (2006) stress that, in this context, building trust with the research team and informing participants in detail is important to convince potential participants. So, we informed the potential participants comprehensively in a separate non-binding event before the start of the course. Besides a presentation, we explained our process in a compact but detailed document, additional to the usual required legal documents. None of the potential participants refused to participate because of privacy or legal considerations.

7.3.2. Data set

Screen Recordings

We developed and installed an application that recorded the participants' tablet screens permanently in the background and transferred the subsequent files to our server. From our screen recording application, we received a total of 1351 video files in MP4 format resulting in a total file size of 179 GB. The videos added up to 167 hours of screen capture, averaging around one hour and eight minutes per day. Our recording app tried to transfer the video

material over the internet to our server. Whenever the tablet was connected via the wifi and not in use, to prevent blocking the internet connection during phases of user activity. Our video quality setting used around 1GB per hour. We reduced the video resolution to half (1024*768) of the display resolution (2048*1536) and recorded with a dynamic frame rate.

LMS data

The LMS provided detailed log files that contained entries about the activities of the users within the system. Moodle exports log files in CSV format, containing nine data fields per entry. The most important fields are the timestamp and the description of the log entry like, "The user with id '16' viewed the course with id '3'," for example. The log files of our nine participants showed an average of 63 log entries per day. We saw a repeated pattern of weekly peaks right before and after the day of the course meetings. When we combined this data with the screen recordings we got per day, we could follow a similar pattern of weekly activity peaks. In total, we collected 19,081 log entries; filtering all admin and teacher-related entries we end up with 11,503 log entries from our participants during the data collection phase.

7.3.3. Experimental Set-up and Pre-processing

We want to compare the duration of sessions from two different data sources, which means that we have to bring the results from both sources into a comparable format.

Online time estimation using screen recordings

The amount of collected video material was too large to analyze the recordings manually for the occurrence of the LMS and compare it to the log files afterward. Some recent research focuses on automated screen recording analysis (Krieter & Breiter, 2018a; Reeves et al., 2019; Frisson et al., 2016). Krieter and Breiter presented an approach for automatically generating log files from mobile screen recordings by using computer vision and machine learning techniques (Krieter & Breiter, 2018a). They show how their approach can be used to generate data for LA independent of the active applications used in the digital learning environment (Krieter & Breiter, 2018b). We use this (Krieter, 2018b) open-source implementation to detect all Moodle activity in the screen recordings and create log files containing log entries for each video frame.

The computer vision and machine learning methods we apply are Tesserakt for optical character recognition(tesseract-ocr, n.d.), OpenCV (OpenCV library, n.d.) for template matching and perception hashes (Buchner, 2018) to find image similarities. Each time the LMS showed up on the user's screen, we summarized these consecutive video frames and saved the start and end of the LMS activity. The aim was to have a format of data that was easy to compare to the results from the Moodle log files.

Online time estimation using Moodle Log Files

As the participants were able to deactivate the screen recordings (in case they felt uncomfortable being recorded in certain situations, for example), for our study we just took those Moodle sessions that were also saved on video. For this reason, we just show results of four of our participants, those that had a sufficient amount of data from both sources to give an example of our approach and methodological contribution. This results in comparing in a total of 140 sessions which we investigate. We processed the Moodle log files multiple times using different thresholds to define our sessions and to estimate the duration. We split the raw log files into per-user log files. The actual content of the log messages is not important in this case. Similarly to the definitions for estimation from the literature, we use different thresholds for closing a session. This time-out value for the inactivity of a user is the most common way to create sessions and calculate the online-time. We generate our sessions with a threshold starting at zero minutes up to 40 minutes, resulting in 41 different versions of session duration. A session must contain at least two entries. Similar to our video logs, a session consists of a timestamp for the start and endpoints.

Comparing the online time estimations

To explore and directly compare the sessions from both data sources, we created dynamic timeline visualizations (see fig. 7.2) using the Google Charts library (Google, n.d.). By doing this we can show how different the results in online time from both data sources are. We tested the 41 variants of the LMS sessions against the sessions from the video material. We estimated the online time for every variant and compared the duration to the one we got from the screen recordings. Based on this we evaluated which session variant works best on a per user basis. The idea was to find the best threshold value (time-out) for the last action that is part of a session. The best value, in this case,

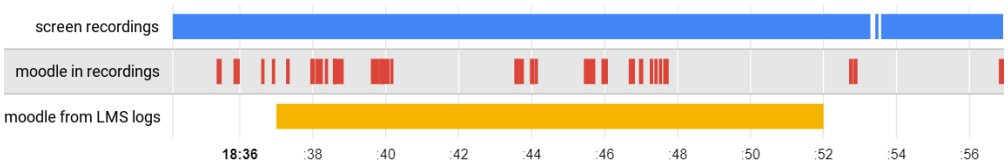

Figure 7.2.: This timeline graphic visualizes an example from our data set. The first line (blue) indicates the recording of the screen. The second line (red) marks the time slots in which we found Moodle activity in the screen recordings. The last line (orange) shows the according session from the Moodle LMS log files.

meant that using this threshold resulted in a session duration that was the closest to the duration we got from the screen recordings of that participant.

7.4. Results

7.4.1. Investigating sessions in time-line visualisations

Several previous research studies in LA which use a session definition of online time estimation are aware of the inaccuracy of the estimation (Kovanović et al., 2015; Sael et al., 2013; Conijn et al., 2017). From our permanent screen recording, we can make direct comparisons between our Moodle logs and what happened at the same time on the screen of our participants. Figure 7.2 shows a timeline visualization of our data sources. In general, we can see that the sessions we got from the LMS logs are clearly different from what we get from the screen recordings. The first line indicates the time spans of the recorded video material (Figure 7.2). The tablet as a mobile device switches off the screen after a period of inactivity or the user actively turns off the screen. In this example, we take a closer look at a time-span of roughly 25 minutes in the evening, starting around 6:35 PM and the screen is active most of the time. The second line shows when the LMS was visible on the screen and the last timeline shows the corresponding session from our Moodle log files. The Moodle log file session, in this case, is specified by the individual threshold value that reflects the online time most accurately for this user (user 1, see the second part of the results section). In this case, this means that the threshold for the last action before we close a session is 30 minutes. This results in an LMS Moodle session of 15 minutes. When we compare the Moodle log file ses-

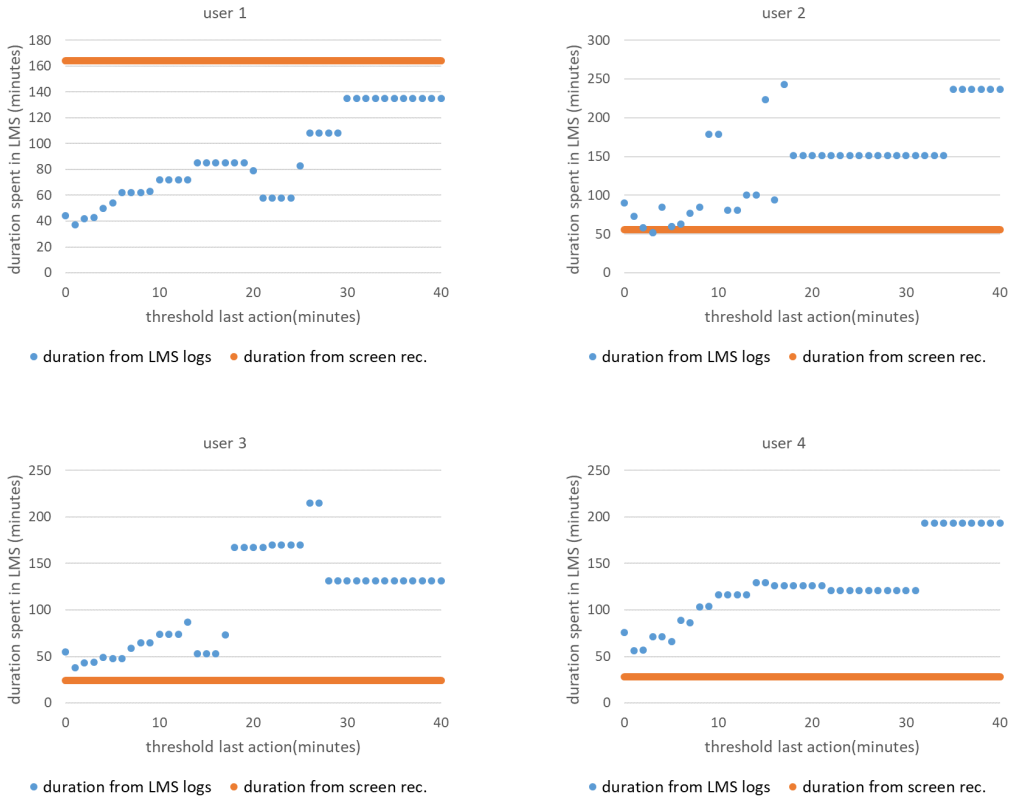

Figure 7.3.: These diagrams show the results from testing timeout thresholds from zero to 40 minutes for the estimation of session duration based on the last action of a user. The blue dots indicate the length of the total online time (y-axis) based on a certain threshold (x-axis). The orange line represents how long Moodle was actually visible on the screen of the user, based on the recordings. For example: for user 1 the LMS was present in the screen recordings for 168 min. (orange line); if we use a threshold of 30 or higher, the results are closest to this value.

sion to the results of Moodle occurrences that we get in the screen recordings, we see that there are many overlaps, but some larger pauses in which Moodle was not on the screen. During the LMS Moodle session, Moodle was on the screen for only four minutes and 18 seconds.

For this particular session, we manually explored the video material in addition to the automated analysis to add some context to the example. A special advantage of the combination of these two data sources is that we get exact information about off-task behavior. We see that Moodle was active on

the participants' screen before the LMS created the first log entry. This results from the user accessing the course website but not yet logging in. Instead, the user creates a bookmark for the LMS and rearranges the pre-configured bookmarks of the tablet's browser. The data of this session is from the beginning of the course. As Moodle can only recognize the user and add a log entry to his or her history when the user is logged in, we have no record of this in the Moodle log files in this case. The first log entries in Moodle are from several attempts of the user to log in, which fails because of trying wrong passwords. The next gap (in comparison to the LMS log session) in the screen recordings is caused by the user changing system preferences. The next few gaps result from switching to the application store several times and searching for specific music applications. The user seems to be unsure which one to install and scrolls through the Google Playstore suggestions for a while and then switches back to the LMS home page. This is followed by rearranging the applications' icons on the Android start screen. The last action of the user is logging actively out of the LMS, which closes the session we get from the Moodle log files. The short occurrence of the LMS in the screen recordings after this results from returning to the Moodle website again, but without logging in again.

7.4.2. Testing different online time estimations

Only ten percent of our sessions end with an active log out click by the user. This means we need an accurate estimation based on the last action in a user-session instead of using log-in and log-out actions. As described in our method section, we use a simplified time-on-session estimation (compared to (Kovanović et al., 2015), for example) to calculate the total time spent using the LMS. We do not take into account specific single time-on-task estimation. The heuristic definition we use to measure the session duration of our Moodle log files is based on estimating the last action of a session based on a time threshold from zero up to 40 minutes. Depending on this definition of sessions, we compare the results of up to 140 sessions.

Figure 7.3 shows the results on the heuristic session duration estimation in relation to the duration the LMS was visible on the screen of the users. The orange line indicates the total session duration (online time in the LMS) from our video recordings. The blue dots specify the total duration spent in the LMS (y-axis), estimated using a threshold t (x-axis). In general, we see some similarities between user two, three, and four.

For user 1 the total duration of the LMS log sessions is closest to the value we get from the screen recordings (168 minutes) for the last action threshold of 30 or higher. We see that a general trend for this user with an ascending threshold is a longer total session duration, despite a drop around the t values 21 until 24. An example session of this user in visualized in Figure 7.2. Although the threshold of 30 seems to result in an imprecise representation for the single example session, overall sessions this threshold leads to the best results for this user. Due to limited space, we cannot show visualizations for all users and sessions.

If we take a look at user 2, we get a different picture. We have a total occurrence of the LMS of 56 minutes in the video data. In this case, a threshold value of two brings our estimation of online time most closely (58 minutes) to the value from the screen recordings. The duration based on other thresholds varies between 52 minutes (t=3) and 237 minutes (t=35 or higher).

For the third user, we had to filter more data as we did not have all LMS log sessions in the screen recordings. This results in 24 minutes of Moodle usage in the screen captures. The closest value to this from our threshold test is for a threshold of one minute, which results in an online-time estimation of 38 minutes based on the Moodle log files. High values for t result in a maximum of 215 minutes of online time (t=26 and 27).

The diagram for user 4 shows an analogous picture, but even the closest total session duration estimation is quite far from the value we get from the screen recordings. In this case, we have 28 minutes of Moodle online time in the screen recordings. The best threshold value in our test is one minute. Using this threshold to calculate the online time results in 56 minutes. The other values also differ strongly from the duration of Moodle occurrence in the screen recordings.

7.4.3. Number of sessions

The focus of this paper is on the duration that students spend online in an LMS. But of course, as we test different session definitions, the number of sessions also changes, not just the duration. Because this number is an important measurement as well, we give short a summary of how the number of sessions changes in connection with changing the time-out threshold for inactivity.

Table 7.2 shows how many sessions there are per user, depending on different values for the time-out of a session in minutes. We chose the values based

Table 7.2.: Number of sessions per user dependent on different time-out thresholds for session estimation.

timeout (min)	User 1	User 2	User 3	User 4
40	18	16	9	21
30	18	16	9	22
15	22	18	13	33
2	30	29	23	33
1	35	38	28	39

on common values used for Moodle log files in the literature (40 minutes, 30 minutes, 15 minutes, see table 7.1). We also added the missing timeouts which resulted in the closest results compared to the screen recordings, which was two for user 3 and user 4 and two minutes for user 2. The number of sessions indicates that based on different timeout thresholds the number of sessions varies. We can see that in general, we get fewer sessions if we increase the time-out, for all users. This could possibly reduce the influence on statistical models, as the values change in a similar ratio for all users.
The data set for the results presented in this section is available online in an anonymized version to make our research transparent and reproducible (Review, 2019).

7.5. Discussion

By comparing the online sessions we get from our Moodle log files to the Moodle activity we get from the screen recordings, we can tell that the actual use of the LMS is very different from what one could expect from just analyzing the LMS log files. To best of our knowledge, there is no previous research data set that can provide the same details on how long and when students actually use the LMS and compare that with parallel LMS log files. By visualizing the data from both sources in parallel timeline graphs we can see that a lot of off-task activity is happening while the user is in an LMS usage session. By adding screen recordings to the data collection and generating log files from the video material, we get a new perspective on the log files. In a way, we can look over the student's shoulder and get an answer to the question, "are you really still using the LMS?". However, just because something is on the user's screen, we

still cannot infer for sure, that the user is actively following or reading the content on the screen. Through the combination of 'classic' log files and screen recordings or log files from screen recordings, we can put our LMS log files in a different context (Hepp et al., 2018). As we did with the description of our example visualization, we can add many insights that we can not infer from Moodle log files. Further research in this direction could automatically analyze how long viewing the content, or an action in Moodle took. Besides that, we could track off-task behavior or on-task behavior that is happening outside of the LMS. In our case, this could be a student reading a task to download a certain music application and use it for an assignment. Using our approach, we could connect these activities and get a different perspective on learning in environments that contain multiple applications.

In previous research in LA, a threshold of 30 or 40 minutes was common for estimating the last action of a session and calculating the online time of a student (e.g. Conijn et al., 2017). In our case, a value of 30 leads to quite exact results for one of our participants (user 1), but not for our three other example cases. If we compare the results of testing thresholds from zero to 40 minutes, we see that the resulting estimated online time varies by more than two hours. As Kovanovic et al. showed, the measurement of how long students stay with a task or the LMS significantly influences the results from statistical models (Kovanović et al., 2015). Therefore, the choice of how to estimate online time is highly important. But from just analyzing our Moodle log files it is not possible to find an exact session duration estimation. For our data set, there is no clear 'winner' threshold that can represent the actual LMS online time in an accurate way. The combination with screen recordings as an additional data source provides context to the traces from the LMS logs and makes it possible to find more accurate thresholds.

The presented sample and analysis of 140 sessions from four participants points in the direction that the 'right' formula for session time estimation is highly individual. In our data, we see very different thresholds that suit the actual time spent, when compared with the screen recordings. Our contribution is to show that existing online-time estimations are not accurate, ie. they over- or underestimate the time. Given the existing definitions, we intend to falsify(Popper, 1963) common assumptions. For this approach, one negative example would be enough, but four are better. Furthermore, we present a new methodological approach (combining log and screen capture data) on how we can to further investigate this. This ought to help a differentiated discussion and deliver first ideas for a novel approach. We are not trying to establish and

prove a new general "perfect" session definition that would undoubtedly require many more participants and data (if even possible). However, for our case, we think a small sample of four participants or 140 sessions is sufficient to prove our point. We show that using several different definitions from the literature, we get very different and inaccurate results and that it seems rather random if they match with the actual online-time estimations we get from the screen recordings. To make this transparent, we make our data and the source code of our approach available to the community.

We strongly encourage further research on how online time estimation of students can be tracked more accurately. We presented a methodological approach to how online time can be determined in an exact and individual way. But the effort to collect screen recording data is significantly higher from a technical perspective, even if we automate the analysis and generate log files from the recordings. From the participants' perspective, recording the screen permanently in the background is privacy-invasive and this can make it hard to find participants willing to join a research study (Krieter, 2019; Reeves et al., 2019; Tang et al., 2006). We see a dilemma here from a research ethical point of view. On the one hand, it deeply affects the privacy of the students to record the screen in the background. On the other hand, it is problematic if predictions are made about the success or failure of students with inaccurately calculated estimations from LMS log files (e.g. Kovanović et al., 2015).

7.5.1. Limitations

We use tablet computers for our data collection. Using other devices like desktop computers, for example, might result in different findings. From the presented study we can not infer an 'ideal' definition of how to estimate the online time of students in an LMS. Additionally, we focused on the threshold variable that decides about the last action of a session. Although this is considered to be the most important factor in estimating online time (see related work section), further research is needed to explore other factors as well (the influence of the number of sessions, for example).

7.6. Conclusions

In this paper, we present an approach to estimate the time students spend in an LMS based on linking LMS logs with parallel log files generated from screen recordings. By this, we can make the actual on- and off-task behavior of students visible. We explore example sessions and visualize the data from both sources in timeline diagrams that indicate great differences between estimation based on Moodle logs and based on screen recordings. We use a common online time estimation strategy from the literature and test different variations in comparison with the results from the screen recordings to find a definition that is most accurate for the individual student. We show that the threshold of minutes of inactivity used to determine the end of a session is critical for calculating the online time and that there are very large deviations. Our findings are in line with the results from previous research on online and on-task time estimations (Sael et al., 2013; Kovanović et al., 2015; Conijn et al., 2017). We showed that the usage sessions we infer from the Moodle log files do not reflect the actual usage time and characteristics that we can observe in the screen recordings. We suggest gathering data in different ways beyond the LMS to overcome the state of having to accept blurry data on online time or time-on-task estimation.

7.6.1. Future work

For future work it would be helpful to make the collection of data on off- and on-task behavior more feasible from a technical perspective for the research team and from a privacy perspective of a participants point of view. Furthermore, by augmenting the LMS log files with the log files from the screen recordings, we can study student behavior in a new and very detailed way and beyond the data points which an LMS can provide. An analysis based on the content on the screen is challenging but highly promising in terms of getting a greater picture of what is happening in a digital learning environment.

Acknowledgment

The authors would like to thank all students for participating in this study. This work was funded by the German Ministry of Education and Research (reference number: 01JKD1709B).

8. Discussion and Conclusion

> **"Which means that if you're reading this now—this sentence—on any sort of modern machine, like a smartphone or tablet, they can follow along and read you."**
> – Snowden (2019)

This quote from Snowden touches on two key aspects of this dissertation. It says that we can track users' actions on mobile devices very closely and understand what a user is doing – great from a research perspective. But in addition to a great source of data to study user behavior, at the same time, we get serious problems and opportunities for surveillance and for abuse of personal privacy.

This chapter reflects on the implications of this work and how it advances the state of the art in the field. As mentioned in the beginning, mobile screens around us are ubiquitous and influence many parts of our everyday life. The way of generating log files presented in this thesis opens new ways for research of looking at the role of mobile devices in people's lives. In short, we developed, evaluated, and applied an approach that shows how we can gather detailed data on mobile devices, independent of which application or needing access to the source code of the OS or these applications.

From here it is still a long way to tell more about the human behavior and intentions behind these data points. We can tell in high detail, what is happing on the screen at any time. But why and for what reason is not easy to infer, if we go beyond easy assumptions like "she is using the camera to send a picture in a chat". The social implications of these interactions are in the context of use and do not happen on the screen of the device alone. In addition, it is important to ask how the privacy of participants in research studies can be taken into account from the beginning in order to facilitate the work of researchers and the involvement of participants in user studies. The next subsection summarizes the contributions of this thesis, followed by a discussion of their implications.

8.1. Contributions

The main contribution of this thesis to the fields of mobile HCI and learning analytics is the introduction of a method for the data collection on mobile devices, based on visual screen output. We contribute to making the mobile device behavior of users accessible, from a perspective, of what is happening on the device's screen at an in-application level of detail. The contributions can be divided into three parts:

- **Automatically analyzing mobile screen recordings**
  This part covers the technical contributions of this work. We show how we can find events automatically in mobile screen recordings to create log files of what is happening on the screen independent of the application. This part focuses on the technical solution of the automatic detection of events in screen recordings of mobile devices and the examination of the impact of different video qualities on the accuracy of the results of the computer vision and machine learning techniques used. In addition, we explore what kind of events can be tracked and how they can be defined. After developing a sever based implementation for analyzing the recordings, we also address the scalability of the approach by developing an on-device version and doing all video processing locally on the mobile device. We show that the approach works reliably on several mobile devices, by conduction long-term test runs.

- **Privacy-friendly**
  Along with the technical solution, this thesis presents the development, implementation, and evaluation of a privacy concept, to expanded usage possibilities of the information-rich but privacy-invasive screen recordings for HCI research. On the one hand, the privacy concept must convince potential participants to take part in research studies involving screen recording and on the other hand to make it easier for researchers to apply this method in user studies. The concept is based on Privacy by Design and involves processing all recordings locally on the users' devices. The contribution can be summarized as showing how to use mobile screen recordings as a long-term data source for log files in a privacy-friendly and a scalable way to analyze the use of mobile applications in detail.

- **Application to generate and analyze data in education**
  In the second part of this dissertation, we introduce the approach as a

new method to collect data in digital learning environments. We use log files from screen recordings as a data source together with log files from an LMS in a learning analytics study. In this context, we can show that by extending common log files with data from screen recordings, measurements of the online-time of students is more precise. Several common approaches from the literature to estimate online time using LMS logs deliver unprecise results for our sample. This part contributes to addressing a common problem when working with log files in learning analytics and mobile HCI research: the data lacks details we can not get from only log file data sources like system log files of a mobile device or from an LMS.

Together these contributions address several challenges we derived from previous work in mobile HCI and learning analytics research on tracking user behavior: generating precise and detailed data of user interactions, the privacy of participants in user studies, and working with rather general log files leading to imprecise estimation of measures (see section 2.3).

This work extends the state of the art in mobile HCI and learning analytics. It greatly advances how we can follow user behavior on mobile devices. Previous work based on mobile log files reported that these are missing details and are rather general. Our approach is not limited by constraints like the need to implement logging commands. Using screen recordings as a basis for log files overcomes this limitation and expands logging capabilities to everything visible on the screen. Previous work involving mobile screen recordings was mostly limited to short periods because the analysis is time-consuming. However, the usefulness of this data source is considered to be of great value. We address this and extend the feasible use of screen recordings to long time spans. Related work on the desktop platform does not take the characteristics of mobile screen recordings into account. We exploit these and created an efficient automatic approach to analyze mobile screen recordings. The approach also proves to be suitable for running locally on the mobile device for analysis of the recordings instead of having to rely on the computing power of a server. Previous research described privacy as a challenge when working with screen recordings in user studies but lacks to address these issues efficiently. The existing solution to this was basically informed consent by the participant. We are considerably expanding privacy efforts related to screen recordings. By processing all recordings locally on the participant's device and only transferring anonymous log files, we present a unique approach to work with screen recordings in a privacy-friendly way. In learning analytics research the focus

of work with LMS log files brings challenges. The need for additional data is being reported as well as issues related to estimating student measurements (e.g. online-time). We show how our approach can generate data in learning environments and how we can use this data to improve the estimation of measurements with high accuracy.

8.2. Implications of the Findings

We spend a lot of our time with screens. For many people working involves at least eight hours of watching a computer screen. Then we spend about one or two hours using our phone during the day and look several times at our smartwatches and finish the day watching TV while we do shopping on an iPad. All these screens can tell a story of our digital footprint. The combination of screen recording analysis and log file generation realizes the merge of two data sources, one usually short-termed and one used for the log-term. This could also be seen as a combination of qualitative and quantitative methods. The manual evaluation of screen recordings is often approached qualitatively, while the work with log data is usually characterized by quantitative methods. In this way, we make it possible to approach screen recordings with quantitative methods. This enables different opportunities to work with mobile screen recordings in research. We can track and log user action inside any application or anything appearing on the screen, without the need to analyze the screen capture manually and after that step run analytics on the generated log files. This enables a new perspective on mobile user behavior. By implementing the processing of screen recordings locally as an application on mobile devices, scaling the approach in large user studies gets easier and makes it unnecessary to transfer large and sensitive screen video data and process it on a central server.

Technically, the presented work advances and exploits how we can work with large amounts of mobile screen recordings to extend the use of this rich data source. Mobile UIs, especially on smartphones, often have a fixed structure, which makes it easy to identify events, because we know where we have to look. By using simple perceptual hashes to compare image regions, we can speed up the process, as we can decide fast and at low cost, if a video frame contains an event of interest before computational more heavy calculations are done on the frame. Another important step to skip redundant frames.

Screen recordings with around 25 FPS contain many frames without anything new happing, as using a UI by a human being is very slow compared to the speed of 25 FPS. This raised the question, if we should record in lower FPS, which we showed, is possible, but at the risk of losing some very short events (described in the first chapter of part one). Regarding privacy in user studies, screen recordings are very different to log files, from the OS, for example. In screen recordings, we can basically see everything, from private family pictures to checking how to quit drug abuse or chatting with an affair. This makes privacy a highly important topic. In most cases to answer a certain research question, we do not need or want to look at all this screen data, but only at specific events that occur. In other words, if we want to study learning processes, we focus on relevant applications and activities related to the topic and are not interested in the student using the device to order a pizza. As a consequence, it is possible to create new HCI data sets for research purposes. Due to the possibility to evaluate screen recordings in a privacy friendly way, the log files from the screen recordings can be passed on for secondary use under certain conditions. This can create a possibility for an anonymous use of screen videos, which was not possible in this form before.s By developing a privacy concept we make a step in the direction that it is less necessary to worry about personal data in user studies using the approach presented in this thesis because less data is leaving the personal device. This makes informing participants easier before consent, as there is less risk of leaking personal data for example. Though as the second publication of this dissertation shows, convincing potential participants remains not easy, if the recording of the screen is involved. This means, even before arriving at the step, at which participants have to sign a legal consent form, it is first of all important to apply a convincing method of data collection and research agenda. In relation to his, in the last paper presented this thesis shows that under certain circumstances, screen recordings are accepted by participants. Mainly by building trust between the research team and participants, explaining the data collection and its goals, and not making the participants having to use their own private mobile devices (see section 7.3.1). Additionally, the research agenda is very different compared to the scenario used in the survey on the privacy concept in chapter five. These results could also refer to the so-called "privacy paradox", which means that people tend to be very concerned about their privacy when asked in surveys, but often behave differently in practice or trade in their data for relatively small rewards (see e.g. the literature review by Kokolakis, 2017). In the case of our survey, this could mean that participants would vote differently if they

had the option to actually participate in a study using the logging approach. As mentioned in the beginning so-called "shoulder surfing" is a problem, because we can easily observe a screen of other people in many places (see e.g. Eiband et al., 2017). This is not only interesting from a privacy perspective but from a security view as well. During the last years, several steps in the development of hard and software address these issues. Safe authentication methods like fingerprint or face detection are ways to work around the need to use a GUI for passwords or unlocking mobile devices. The mechanisms also do protect these secrets if the screen is being recorded or watched by someone else. Another direction is to go for local solutions on the mobile device. It is a strong point currently in trying to convince and get trusted by customers or users, that their data is not leaving their device. Google, for example, announced to do all face detection steps locally on their new Pixel device (see section 5.7). Another privacy-related implication are abuse scenarios of surveillance this approach enables. By making changes to the developed application which processes the recordings on mobile devices (which is open source), it is possible to hide the collection of data from the user. This opens up the possibility of spying on users unnoticed. There are other ways to secretly spy on mobile devices, not only via screen recordings. Nevertheless, the approach presented is another technical variation on how data can be collected illegally for unpredictable purposes. This is a general issue which needs further attention in digital research.

The second part of this thesis gives an example of how the presented logging approach can be used in a research study to track the online behavior of students. In this case in the context of a study in a setting of learning using tablet devices. Evaluating the results of LMS log file measurements by comparing the results with log files from screen recordings opens new ways of evaluating measurements and data-preprocessing for learning analytics. This contribution advances both fields, HCI, and learning analytics. Just collecting the data is not enough. We showed how we can get almost any event happening in the recordings at a low computational cost but to make sense of this data and put it into context and make means of it is another step. For our example from learning analytics, by automatically analyzing the recordings and analyzing the LMS log files, we show that the LMS online time we can measure from both data sources differs significantly. The calculation of the time spent by students in the LMS, which in some cases varies greatly, clearly shows how great the influence of decisions is made during data preparation by the researchers. A different or modified definition of a construct such as

the length of time spent in the LMS can lead to significantly different results in later statistical models for predictions. By this example, we can show how using this method can extend existing data sources and help to get more accurate research data, in this case, estimation of the online time of the students. This overcomes a limitation of log files from Learning Management Systems, namely that we do not receive any information about what happens outside the LMS and whether the LMS is actually being used as expected from the LMS log files. From a learning analytics perspective, a more detailed analysis of what is happing inside and outside of the LMS from the screen recordings would be the next step. Further research could go in the direction of tracking on and off-task behavior of students or connecting the analysis of contents viewed within the LMS in actions outside in different application which are related to this. We showed that students are switching between the use of the LMS to other activities. We could see this as a first step, what we can do with this more detailed data from the recordings. But another step from here would be to make sense of these usage breaks and analyze what is happening in these breaks and why. The students could still engage with something else related to the task, whether digital or analog. As other previous findings suggest, just the data points from an LMS do not tell the whole story. This raises the question of what we actually measure and how and if these measures are reliable for their purpose. Machines and calculations are often perceived as precise and correct by humans because it is basically "math". This makes it important to question the processes, measures, and calculations we are applying, not only in learning analytics.

8.3. Limitations

The approach presented brings advances in how we can track interactions on mobile devices, but this also brings limitations imposed by the research design and methods chosen.

Reflecting on the limitations of the single chapters, an overall limitation of the presented approach is that we can "only" interpret what is happening on the screen. But we can not say for sure if the user is really paying attention to what is happening on the screen for example. For our estimation of the online time of students, this means that although we know exactly for how long the LMS hat been present on the screen, this does not mean the students were

really interacting with the system or paying attention. Also, we do not know if multiple users are sharing something in a conversation using one device. These contexts are not easy to infer from only log file analysis without adding other sources of data. Through the way of logging presented in this thesis, we can generate data on mobile devices at a novel and high detailed level. A major issue is the analysis of this data. Collecting all this data seems like the easier part: making sense of all these data points in a meaningful way remains a current challenge.

In regard to the user study on the privacy concept, it seems promising to do research on how incentives (e.g. money, coupons) may have changed the agreement rate on a specific research scenario and method of collecting data. This was not part of the evaluation, which limits its significance for further studies working with incentives. Another limitation in relation to the evaluation and method used to do this is the possible influence of the previously mentioned privacy paradox, which might have an effect as well. The results could be different if the participants really had to join the research study instead of just presenting different scenarios to choose from. Besides that this study was limited by the number of participants, which was 35. Overall, the privacy aspect of mobile screen recordings remains a challenge in user studies, as the evaluation of the implemented privacy concept showed. This result might be biased, if we make the assumption that german students in computer science might be more aware and critical regarding privacy issues, compared to other groups. Nevertheless recording the screen is privacy-invasive in its nature. The implementation and evaluation of our privacy concept points in the direction, that even when addressing privacy carefully, permanent recording of the screen remains problematic to some extend.

From our study on the LMS online-time estimation of students, we can demonstrate, that common measures for online-time do not work precisely for our students. But we cannot infer a better general definition of how to infer the online-time from just LMS data. Our sample was too small and too diverse to go in this direction. And even if we are much more precise by estimating the online-time by using screen recordings, one could argue that we still had some decision to take during the process of analysis which influences the estimation results. For example, the LMS may be present on the screen, but only the login page, without the user interacting with it, which probably could be considered as not being online in the LMS.

From a rather technical perspective, some more limitations apply. Although most of the work and code in this thesis is related to Android, the

way of generating logs from screen recordings works for videos from iOS as well. Besides that, updates of the UI of applications or the OS can make updating the event definitions necessary as they are based on GUI states. The same holds true for differences in the UI of different Android versions. Our evaluation of our method in the first chapter is based on a dataset of 118 screen recordings and 30 test events. We can say how successful our approach works for these events and this data set, but this does not mean using this approach with different events and other screen recording data will result in the same precise rate of event detection. This limits our approach, for example, compared to system log files, which are less error-prone.

8.4. Future Work

The research approach presented, greatly advances how we can track interactions on mobile devices, but also triggers new questions. We have to further develop methods and suited analytics to enhance research on user behavior in relation to the digital devices surrounding us. Adding additional data at a new level of detail compared to other sources does not solve the question of how we can effectively use this data to answer more than just questions from a technical perspective.

In terms of further developing and enhancing this way of generating log data, there are several starting points. A "live" detection of events would be interesting to trigger questions or surveys to the participants, based on their recent actions. In terms of performance, there are several starting points: expanding multi-core usage or finding faster ways to detect redundant frames. In terms of privacy, it would be interesting to work with features like the possibility to check or uncheck events from a list for participants to control which data they agree on collecting in a study. In general, it would be valuable to run studies with larger sample size and for long-terms applying the presented method, to further enhance the technical solution from these experiences.

We are using methods of computer vision on the video frames and methods of machine learning by applying tesseract for OCR. Another future approach could be to use the current approach to label video frames and use these labeled recordings to train image or video classifiers. Then these could help to create a more general detection of events that are not sensitive for small changes of GUIs for example. As this approach would rely on probabilities and

is less easy to debug, compared to the computer vision and OCR techniques of our approach, this could make the log data more error-prone.

We addressed one specific estimation of a measurement which is important in the pre-processing of log data in learning analytics. Expanding the sample size and gathering more data on the online-time calculation would be a valuable research objective. Especially the privacy-friendly on-device processing of screen recordings could help to run a similar study at large-scale. This might lead to a new "gold" standard if backed by extensive data from screen recordings and LMS data. For future work, it would be interesting to do this with other steps in pre-processing as well, as the number of sessions. Also, the time-on-task and time-off-task behavior of students would be an interesting topic for further research, exploiting possibilities of detailed data from logs generated from screen recordings on the breaks of LMS usage, for example. Furthermore, the method presented can be used to collect data on learning processes outside the LMS in digital learning environments. Several previous and related works support the idea of using other data sources in learning analytics. One example is the study presented in chapter seven, in which we collected data on tablet devices in music courses. In this constellation, additional data sources are of great value, since LMS log files cannot provide data on learning processes in music applications. However, this is true for many other learning scenarios where our approach could help to generate valuable learning data that we have not been able to capture in this way before.
During this dissertation project, new approaches and implementations of learning analytics emerged. It is controversial, which data we should collect and how we should use this data. Since we are significantly expanding the logging capabilities on mobile devices, this is an important issue. A recent example from learning analytics is the "Spotter" application (Spotter, 2020) which is in use at a couple of Universities (Drew, 2019). This tool uses techniques of collecting data using Bluetooth beacons and Wifi which are also in use in order to trace back contact of people during pandemics for example (e.g. during the corona pandemic). These are two fundamentally different objectives: the attendance of students and the containment of a virus. This demonstrates one of the issues that need further discussion – not only in learning analytics research – the balance between what we win by collecting and analyzing certain data on the one hand and what risk for privacy and surveillance we add up on the other hand.

8.5. Conclusion

The overall goal of this dissertation is to find a way to collect data for analyzing user behavior at high detail, regardless of which application, feasible for long-terms, and privacy-friendly on mobile devices.

We achieve this goal by combining the strength of two common data sources, screen recordings and log files. Screen recordings fulfill the goal of providing high details of user activity and being independent of applications. Log files are known for long-term analysis and are suitable for the protection of user privacy.

This work greatly advances how we can follow user behavior on mobile devices. The contributions of this work open a new perspective of research possibilities of computer science in the fields of mobile HCI and learning analytics in a threefold way. We can generate long-term log data at high detail from mobile screen recordings, we can do it privacy-friendly and easy to scale on mobile devices. Additionally, we show how to successfully apply this method in a learning analytics context and show how we can give a new perspective on how to improve the estimation of student performance measures.

References

Agudo-Peregrina, n. F., Iglesias-Pradas, S., Conde-González, M. n., & Hernández-García, n. (2014, February). Can we predict success from log data in VLEs? Classification of interactions for learning analytics and their relation with performance in VLE-supported F2f and online learning. Computers in Human Behavior, 31, 542–550. Retrieved from http://www.sciencedirect.com/science/article/pii/S074756321300188X doi: 10.1016/j.chb.2013.05.031

Aho, P., Kanstrén, T., Räty, T., & Röning, J. (2014). Automated extraction of gui models for testing. In Advances in computers (Vol. 95, pp. 49–112). Elsevier.

Aho, P., Kanstren, T., Räty, T., & Röning, J. (2014). Advances in Computers (A. Memon, Ed.). Academic Press.

Ayalon, O., Toch, E., Hadar, I., & Birnhack, M. (2017). How Developers Make Design Decisions About Users' Privacy: The Place of Professional Communities and Organizational Climate. In Companion of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing (pp. 135–138). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/3022198.3026326 doi: 10.1145/3022198.3026326

Balagtas-Fernandez, F., & Hussmann, H. (2009). A Methodology and Framework to Simplify Usability Analysis of Mobile Applications. In Proceedings of the 2009 IEEE/ACM International Conference on Automated Software Engineering (pp. 520–524). Washington, DC, USA: IEEE Computer Society. Retrieved 2017-04-28, from http://dx.doi.org/10.1109/ASE.2009.12 doi: 10.1109/ASE.2009.12

Ba-Omar, H., Petrounias, I., & Anwar, F. (2007, July). A Framework for Using Web Usage Mining to Personalise E-learning. In (pp. 937–938). IEEE. Retrieved 2019-08-01, from http://ieeexplore.ieee.org/document/4281208/ doi: 10.1109/ICALT.2007.13

Barbello, B. (2019, July). (Don't) hold the phone: new features coming to Pixel 4. Retrieved from https://www.blog.google/products/pixel/new-features-pixel4/

Barkuus, L., & Dey, A. (2003). Location-Based Services for Mobile Telephony: a Study of Users' Privacy Concerns. In In 9th Ifip Tc13 International Conference on Human-Computer Interaction, Interact 2003.

Barnum, C. M. (2010). Usability Testing Essentials: Ready, Set...Test! Elsevier.

Böhmer, M., Hecht, B., Schöning, J., Krüger, A., & Bauer, G. (2011). 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 (pp. 47–56). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2037373.2037383 doi: 10.1145/2037373.2037383

Blikstein, P. (2011). Using Learning Analytics to Assess Students' Behavior in Open-ended Programming Tasks. In Proceedings of the 1st International Conference on Learning Analytics and Knowledge (pp. 110–116). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2090116.2090132 (event-place: Banff, Alberta, Canada) doi: 10.1145/2090116.2090132

Boker, S. M., Brick, T. R., Pritikin, J. N., Wang, Y., Oertzen, T. v., Brown, D., ... Neale, M. C. (2015, November). Maintained Individual Data Distributed Likelihood Estimation (MIDDLE). Multivariate Behavioral Research, 50(6), 706–720. Retrieved 2019-07-24, from https://doi.org/10.1080/00273171.2015.1094387 doi: 10.1080/00273171.2015.1094387

Borycki, E. M., Monkman, H., Griffith, J., & Kushniruk, A. W. (2015). Mobile usability testing in healthcare: Methodological approaches. In Medinfo (pp. 338–342).

Brown, B., McGregor, M., & McMillan, D. (2014). 100 Days of iPhone Use: Understanding the Details of Mobile Device Use. In Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services (pp. 223–232). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2628363.2628377 doi: 10.1145/2628363.2628377

Brown, B., Weilenmann, A., McMillan, D., & Lampinen, A. (2016). Five Provocations for Ethical HCI Research. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (pp. 852–863). New York, NY,

USA: ACM. Retrieved from http://doi.acm.org/10.1145/2858036.2858313 doi: 10.1145/2858036.2858313

Buchner, J. (2018, January). imagehash: A Python Perceptual Image Hashing Module. Retrieved from https://github.com/JohannesBuchner/imagehash (original-date: 2013-03-02T23:32:48Z)

Canny, J. (1986, November). A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8(6), 679–698. doi: 10.1109/TPAMI.1986.4767851

Carroll, J. M. (1997, April). Human-computer interaction. International Journal of Human-Computer Studies, 46(4), 501–522. Retrieved 2020-04-23, from https://doi.org/10.1006/ijhc.1996.0101 doi: 10.1006/ijhc.1996.0101

Cavoukian, A. (2011). Privacy by Design - The 7 Foundational Principles.

Chang, T.-H., Yeh, T., & Miller, R. C. (2010). GUI Testing Using Computer Vision. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 1535–1544). New York, NY, USA: ACM. Retrieved 2017-04-19, from http://doi.acm.org/10.1145/1753326.1753555 doi: 10.1145/1753326.1753555

Conijn, R., Snijders, C., Kleingeld, A., & Matzat, U. (2017, January). Predicting Student Performance from LMS Data: A Comparison of 17 Blended Courses Using Moodle LMS. IEEE Transactions on Learning Technologies, 10(1), 17–29. doi: 10.1109/TLT.2016.2616312

Cooper, A., et al. (2012). What is analytics? definition and essential characteristics. CETIS Analytics Series, 1(5), 1–10.

Damianov, D., Kupczynski, L., Calafiore, P., Damianova, E., Soydemir, G., & Gonzalez, E. (2009, January). Time Spent Online and Student Performance in Online Business Courses: A Multinomial Logit Analysis1.

Dawson, S., Mcwilliam, E., & Tan, J. P.-L. (2008). Teaching smarter: How mining ICT data can inform and improve learning..

del Valle, R., & Duffy, T. M. (2009, March). Online learning: Learner characteristics and their approaches to managing learning. Instructional Science, 37(2), 129–149. Retrieved 2019-09-23, from https://doi.org/10.1007/s11251-007-9039-0 doi: 10.1007/s11251-007-9039-0

Do, T. M. T., Blom, J., & Gatica-Perez, D. (2011). Smartphone Usage in the Wild: A Large-scale Analysis of Applications and Context. In Proceedings

of the 13th International Conference on Multimodal Interfaces (pp. 353–360). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2070481.2070550 doi: 10.1145/2070481.2070550

Drachsler, H., & Greller, W. (2012). The Pulse of Learning Analytics Understandings and Expectations from the Stakeholders. In Proceedings of the 2nd International Conference on Learning Analytics and Knowledge (pp. 120–129). New York, NY, USA: ACM. Retrieved 2019-11-21, from http://doi.acm.org/10.1145/2330601.2330634 (event-place: Vancouver, British Columbia, Canada) doi: 10.1145/2330601.2330634

Drew, H. (2019, December). Colleges are turning students' phones into surveillance machines, tracking the locations of hundreds of thousands. Washington Post. Retrieved 2020-04-01, from https://www.washingtonpost.com/technology/2019/12/24/colleges-are-turning-students-phones-into-surveillance-machines-tracking-locations-hundreds-thousands/ (Library Catalog: www.washingtonpost.com)

Dyment, J., Stone, C., & Milthorpe, N. (2020, March). Beyond busy work: rethinking the measurement of online student engagement. Higher Education Research & Development, 0(0), 1–14. Retrieved 2020-03-24, from https://doi.org/10.1080/07294360.2020.1732879 (Publisher: Routledge _eprint: https://doi.org/10.1080/07294360.2020.1732879) doi: 10.1080/07294360.2020.1732879

Ebert, A., Gershon, N. D., & Veer, G. C. v. d. (2012, May). Human-Computer Interaction. KI - Künstliche Intelligenz, 26(2), 121–126. Retrieved 2017-08-09, from https://link.springer.com/article/10.1007/s13218-012-0174-7 doi: 10.1007/s13218-012-0174-7

Eiband, M., Khamis, M., von Zezschwitz, E., Hussmann, H., & Alt, F. (2017, May). Understanding Shoulder Surfing in the Wild: Stories from Users and Observers. In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems (pp. 4254–4265). Denver, Colorado, USA: Association for Computing Machinery. Retrieved 2020-03-31, from https://doi.org/10.1145/3025453.3025636 doi: 10.1145/3025453.3025636

Eick, S. G., Nelson, M. C., & Schmidt, J. D. (1994, December). Graphical Analysis of Computer Log Files. Commun. ACM, 37(12), 50–56. Retrieved from http://doi.acm.org/10.1145/198366.198378 doi: 10.1145/198366.198378

Ernala, S. K., Burke, M., Leavitt, A., & Ellison, N. B. (2020). How well do people report time spent on facebook? an evaluation of established survey questions with recommendations. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/1753326.1753555 doi: 10.1145/1753326.1753555

Fast, K. (2002, August). Recording Screen Activity During Usability Testing (Tech. Rep.). Retrieved 2020-04-27, from https://boxesandarrows.com/recording-screen-activity-during-usability-testing/ (Library Catalog: boxesandarrows.com Section: Discovery, Research, and Testing)

Fernandez-Medina, C., Pérez-Pérez, J. R., Álvarez García, V. M., & Paule-Ruiz, M. d. P. (2013). Assistance in Computer Programming Learning Using Educational Data Mining and Learning Analytics. In Proceedings of the 18th ACM Conference on Innovation and Technology in Computer Science Education (pp. 237–242). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2462476.2462496 (event-place: Canterbury, England, UK) doi: 10.1145/2462476.2462496

Ferreira, D., Kostakos, V., & Dey, A. K. (2015). AWARE: Mobile Context Instrumentation Framework. Frontiers in ICT, 2. Retrieved 2018-11-29, from https://www.frontiersin.org/articles/10.3389/fict.2015.00006/full doi: 10.3389/fict.2015.00006

Frisson, C., Malacria, S., Bailly, G., & Dutoit, T. (2016). InspectorWidget: A System to Analyze Users Behaviors in Their Applications. In Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems (pp. 1548–1554). New York, NY, USA: ACM. Retrieved 2017-04-13, from http://doi.acm.org/10.1145/2851581.2892388 doi: 10.1145/2851581.2892388

Gašević, D., Dawson, S., & Siemens, G. (2015, January). Let's not forget: Learning analytics are about learning. TechTrends, 59(1), 64–71. Retrieved 2018-03-26, from https://link.springer.com/article/10.1007/s11528-014-0822-x doi: 10.1007/s11528-014-0822-x

Gebel, T., Grenzer, M., Kreusch, J., Liebig, S., Schuster, H., Tscherwinka, R., ... Witzel, A. (2015). Verboten ist, was nicht ausdrücklich erlaubt ist: Datenschutz in qualitativen Interviews. Forum: Qualitative Sozialforschung / Forum: Qualitative Social Research, 16(2). Retrieved 2017-07-25, from https://pub.uni-bielefeld.de/publication/2735504

Google. (n.d.). Google Charts library. Retrieved 2019-06-06, from https://developers.google.com/chart/interactive/docs/gallery/timeline

Gürses, S., Troncoso, C., & Diaz, C. (2011). Engineering privacy by design. Computers, Privacy & Data Protection, 14(3), 25.

Haak, M. v. d., Jong, M. D., & Schellens, P. J. (2003, September). Retrospective vs. concurrent think-aloud protocols: Testing the usability of an online library catalogue. Behaviour & Information Technology, 22(5), 339–351. Retrieved from http://dx.doi.org/10.1080/0044929031000 doi: 10.1080/0044929031000

Hepp, A., Breiter, A., & Friemel, T. N. (2018). Digital traces in context| digital traces in context – an introduction. International Journal of Communication, 12, 11.

Hong, J. I., Boriello, G., L, J. A., Mcdonald, D. W., Schilit, B. N., & Tygar, J. D. (2003). Privacy and security in the locationenhanced world wide web. In In Proceedings of Fifth International Conference on Ubiquitous Computing: Ubicomp 2003 (Workshop on Ubicomp Communities: Privacy as Boundary Negotiation (p. 2003).

Hong, J. I., & Landay, J. A. (2004). An Architecture for Privacy-sensitive Ubiquitous Computing. In Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services (pp. 177–189). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/990064.990087 doi: 10.1145/990064.990087

IDC, & Gartner. (2018). Share of android os of global smartphone shipments from 1st quarter 2011 to 1st quarter 2018. Statista - The Statistics Portal. Retrieved from https://www.statista.com/statistics/236027/global-smartphone-os-market-share-of-android/

Ifenthaler, D., & Schumacher, C. (2016, October). Student perceptions of privacy principles for learning analytics. Educational Technology Research and Development, 64(5), 923–938. Retrieved 2019-11-21, from https://doi.org/10.1007/s11423-016-9477-y doi: 10.1007/s11423-016-9477-y

Imler, B., & Eichelberger, M. (2011, January). Using screen capture to study user research behavior. Library Hi Tech, 29(3), 446–454. Retrieved 2020-01-22, from https://doi.org/10.1108/07378831111174413 doi: 10.1108/07378831111174413

Jansen, B. J., Taksa, I., & Spink, A. (2008). Handbook of Research on Web Log Analysis (B. J. Jansen, Ed.). IGI Global.

Kawalek, J., Stark, A., & Riebeck, M. (2008, February). A New Approach to Analyze Human-mobile Computer Interaction. J. Usability Studies, 3(2), 90–98. Retrieved 2017-04-13, from http://dl.acm.org/citation.cfm?id=2835562.2835566

Kitto, K., Cross, S., Waters, Z., & Lupton, M. (2015). Learning Analytics Beyond the LMS: The Connected Learning Analytics Toolkit. In Proceedings of the Fifth International Conference on Learning Analytics And Knowledge (pp. 11–15). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2723576.2723627 doi: 10.1145/2723576.2723627

Kokolakis, S. (2017, January). Privacy attitudes and privacy behaviour: A review of current research on the privacy paradox phenomenon. Computers & Security, 64, 122–134. Retrieved 2020-03-17, from http://www.sciencedirect.com/science/article/pii/S0167404815001017 doi: 10.1016/j.cose.2015.07.002

Kovanović, V., Gašević, D., Dawson, S., Joksimović, S., Baker, R. S., & Hatala, M. (2015). Penetrating the Black Box of Time-on-task Estimation. In Proceedings of the Fifth International Conference on Learning Analytics And Knowledge (pp. 184–193). New York, NY, USA: ACM. Retrieved 2019-09-11, from http://doi.acm.org/10.1145/2723576.2723623 (event-place: Poughkeepsie, New York) doi: 10.1145/2723576.2723623

Krieter, P. (2018a, 22). Android background screen recorder. (https://github.com/pkrieter/android-background-screen-recorder/)

Krieter, P. (2018b, 22). Mobile screen recordings to log file. (https://github.com/pkrieter/mobile-screen-recordings-to-log-file/)

Krieter, P. (2018c). Project source code screenlogger. Retrieved 2019-10-10, from https://github.com/pkrieter/screenlogger-android

Krieter, P. (2019). Can I Record Your Screen?: Mobile Screen Recordings As a Long-term Data Source for User Studies. In Proceedings of the 18th International Conference on Mobile and Ubiquitous Multimedia (pp. 23:1–23:10). New York, NY, USA: ACM. Retrieved 2019-12-11, from http://doi.acm.org/10.1145/3365610.3365618 (event-place: Pisa, Italy) doi: 10.1145/3365610.3365618

Krieter, P., & Breiter, A. (2018a). Analyzing Mobile Application Usage: Generating Log Files from Mobile Screen Recordings. In Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services (pp. 9:1–9:10). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/3229434.3229450 doi: 10.1145/3229434.3229450

Krieter, P., & Breiter, A. (2018b). Track every move of your students: log files for Learning Analytics from mobile screen recordings. Gesellschaft für Informatik e.V. Retrieved 2019-12-11, from http://dl.gi.de/handle/20.500.12116/21042

Kushniruk, A. W., & Borycki, E. M. (2006). Low-Cost Rapid Usability Engineering: Designing and Customizing Usable Healthcare Information Systems. Retrieved 2020-04-28, from https://dspace.library.uvic.ca//handle/1828/6380 (Accepted: 2015-07-24T20:49:15Z Publisher: Longwoods Publishing)

Kushniruk, A. W., & Patel, V. L. (2004, February). Cognitive and usability engineering methods for the evaluation of clinical information systems. Journal of Biomedical Informatics, 37(1), 56–76. Retrieved from http://www.sciencedirect.com/science/article/pii/S1532046404000206 doi: 10.1016/j.jbi.2004.01.003

Lafreniere, B., Grossman, T., Matejka, J., & Fitzmaurice, G. (2014, April). Investigating the feasibility of extracting tool demonstrations from in-situ video content. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 4007–4016). Toronto, Ontario, Canada: Association for Computing Machinery. Retrieved 2020-04-28, from https://doi.org/10.1145/2556288.2557142 doi: 10.1145/2556288.2557142

LAK-Conference. (2011). 1st International Conference on Learning Analytics and Knowledge 2011 | Connecting the technical, pedagogical, and social dimensions of learning analytics. Retrieved 2020-02-05, from https://tekri.athabascau.ca/analytics/

Langheinrich, M. (2001). Privacy by Design — Principles of Privacy-Aware Ubiquitous Systems. In G. D. Abowd, B. Brumitt, & S. Shafer (Eds.), Ubicomp 2001: Ubiquitous Computing (pp. 273–291). Springer Berlin Heidelberg.

Lazar, J., Feng, J. H., & Hochheiser, H. (2017). Research Methods in Human-Computer Interaction. Morgan Kaufmann. (Google-Books-ID: hbkxDQAAQBAJ)

Lettner, F., & Holzmann, C. (2012a). Automated and Unsupervised User Interaction Logging As Basis for Usability Evaluation of Mobile Applications. In Proceedings of the 10th International Conference on Advances in Mobile Computing & Multimedia (pp. 118–127). New York, NY, USA: ACM. Retrieved 2017-04-19, from http://doi.acm.org/10.1145/2428955.2428983 doi: 10.1145/2428955.2428983

Lettner, F., & Holzmann, C. (2012b, March). Sensing mobile phone interaction in the field. In 2012 IEEE International Conference on Pervasive Computing and Communications Workshops (pp. 877–882). doi: 10.1109/PerComW.2012.6197635

Liang, H., Song, H., Fu, Y., Cai, X., & Zhang, Z. (2011, June). A remote usability testing platform for mobile phones. In 2011 IEEE International Conference on Computer Science and Automation Engineering (Vol. 2, pp. 312–316). doi: 10.1109/CSAE.2011.5952477

lookback.io. (2020). Lookback: Simple and powerful user research. Retrieved 2020-04-01, from https://lookback.io/

Lowrance, W. (2003, July). Learning from experience: privacy and the secondary use of data in health research. Journal of Health Services Research & Policy, 8(1_suppl), 2–7. Retrieved from http://dx.doi.org/10.1258/135581903766468800 doi: 10.1258/135581903766468800

Marquardt, C. G., Becker, K., & Ruiz, D. D. (2004, July). A pre-processing tool for Web usage mining in the distance education domain. In Proceedings. International Database Engineering and Applications Symposium, 2004. IDEAS '04. (pp. 78–87). doi: 10.1109/IDEAS.2004.1319780

Matejka, J., Grossman, T., & Fitzmaurice, G. (2013). Patina: Dynamic Heatmaps for Visualizing Application Usage. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (pp. 3227–3236). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2470654.2466442 doi: 10.1145/2470654.2466442

McMillan, D., McGregor, M., & Brown, B. (2015). From in the Wild to in Vivo: Video Analysis of Mobile Device Use. In Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services (pp. 494–503). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2785830.2785883 doi: 10.1145/2785830.2785883

McMillan, D., Morrison, A., Brown, O., Hall, M., & Chalmers, M. (2010, January). Further into the Wild: Running Worldwide Trials of Mobile Systems. In (Vol. 6030, pp. 210–227). doi: 10.1007/978-3-642-12654-3_13

Munk, M., & Drlík, M. (2011, January). Impact of Different Pre-Processing Tasks on Effective Identification of Users' Behavioral Patterns in Web-based Educational System. Procedia Computer Science, 4, 1640–1649. Retrieved 2019-09-12, from http://www.sciencedirect.com/science/article/pii/S1877050911002353 doi: 10.1016/j.procs.2011.04.177

Oliner, A., Ganapathi, A., & Xu, W. (2012, February). Advances and Challenges in Log Analysis. Commun. ACM, 55(2), 55–61. Retrieved from http://doi.acm.org/10.1145/2076450.2076466 doi: 10.1145/2076450.2076466

OpenCV library. (n.d.). Retrieved 2018-01-12, from https://opencv.org/

Papamitsiou, C., & Economides, A. (2014, October). Learning Analytics and Educational Data Mining in Practice: A Systematic Literature Review of Empirical Evidence. Educational Technology & Society, 17, 49–64.

Pardo, A., & Kloos, C. D. (2011). Stepping out of the Box: Towards Analytics Outside the Learning Management System. In Proceedings of the 1st International Conference on Learning Analytics and Knowledge (pp. 163–167). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2090116.2090142 doi: 10.1145/2090116.2090142

Planas, E., & Cabot, J. (2020). How are UML class diagrams built in practice? A usability study of two UML tools: Magicdraw and Papyrus. Computer Standards and Interfaces, 67. doi: 10.1016/j.csi.2019.103363

Popper, K. R. (1963). Science as falsification. Conjectures and refutations, 1, 33–39.

Ramler, R., Gattringer, M., & Pichler, J. (2020, February). Live Replay of Screen Videos: Automatically Executing Real Applications as Shown in Recordings. In 2020 IEEE 27th International Conference on Software Analysis, Evolution and Reengineering (SANER) (pp. 664–665). (ISSN: 1534-5351) doi: 10.1109/SANER48275.2020.9054833

Reeves, B., Ram, N., Robinson, T. N., Cummings, J. J., Giles, C. L., Pan, J., ... Yeykelis, L. (2019, March). Screenomics: A Framework to Capture and Analyze Personal Life Experiences and the Ways that Technology Shapes Them.

Human–Computer Interaction, 0(0), 1–52. Retrieved from https://doi.org/10.1080/07370024.2019.1578652 doi: 10.1080/07370024.2019.1578652

Review, B. (2019, 22). Removed for blind review. (https://url-to-data-set.com/)

Ribeiro, J., Saghezchi, F. B., Mantas, G., Rodriguez, J., Shepherd, S. J., & Abd-Alhameed, R. A. (2020, February). An Autonomous Host-Based Intrusion Detection System for Android Mobile Devices. Mobile Networks and Applications, 25(1), 164–172. Retrieved 2020-03-25, from https://doi.org/10.1007/s11036-019-01220-y doi: 10.1007/s11036-019-01220-y

Romero, C., & Ventura, S. (2013). Data mining in education. WIREs Data Mining and Knowledge Discovery, 3(1), 12–27. Retrieved 2020-01-27, from https://onlinelibrary.wiley.com/doi/abs/10.1002/widm.1075 doi: 10.1002/widm.1075

Rubenstein, I., & Good, N. (2013, December). Privacy by Design: A Counterfactual Analysis of Google and Facebook Privacy Incidents. Berkeley Technology Law Journal, 28(2). Retrieved from http://scholarship.law.berkeley.edu/btlj/vol28/iss2/6 doi: doi:10.15779/Z38G11N

Ruhleder, K., & Jordan, B. (1997). Capturing Complex, Distributed Activities: Video-Based Interaction Analysis as a Component of Workplace Ethnography. In A. S. Lee, J. Liebenau, & J. I. DeGross (Eds.), Information Systems and Qualitative Research: Proceedings of the IFIP TC8 WG 8.2 International Conference on Information Systems and Qualitative Research, 31st May–3rd June 1997, Philadelphia, Pennsylvania, USA (pp. 246–275). Boston, MA: Springer US. Retrieved 2020-04-27, from https://doi.org/10.1007/978-0-387-35309-8_14 doi: 10.1007/978-0-387-35309-8_14

Sael, N., Marzak, A., & Behja, H. (2013, May). Web Usage Mining data preprocessing and multi level analysis on Moodle. In 2013 ACS International Conference on Computer Systems and Applications (AICCSA) (pp. 1–7). doi: 10.1109/AICCSA.2013.6616427

Schaar, P. (2010, August). Privacy by Design. Identity in the Information Society, 3(2), 267–274. Retrieved 2017-07-17, from https://link.springer.com/article/10.1007/s12394-010-0055-x doi: 10.1007/s12394-010-0055-x

Sears, A., & Jacko, J. A. (2009). Human-Computer Interaction: Designing for Diverse Users and Domains. CRC Press.

Siemens, G. (2013, October). Learning Analytics: The Emergence of a Discipline. American Behavioral Scientist, 57(10), 1380–1400. Retrieved from https://doi.org/10.1177/0002764213498851 doi: 10.1177/0002764213498851

Snowden, E. (2019). Permanent Record. Pan Macmillan. Retrieved from https://books.google.de/books?id=9aamDwAAQBAJ

Spiekermann, S., & Cranor, L. F. (2009, January). Engineering Privacy. IEEE Transactions on Software Engineering, 35(1), 67–82. doi: 10.1109/TSE.2008.88

Spotter. (2020). Retrieved 2020-04-01, from https://spotteredu.com (Library Catalog: spotteredu.com)

Tang, J. C., Liu, S. B., Muller, M., Lin, J., & Drews, C. (2006). Unobtrusive but Invasive: Using Screen Recording to Collect Field Data on Computer-mediated Interaction. In Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work (pp. 479–482). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/1180875.1180948 doi: 10.1145/1180875.1180948

Tempelaar, D. T., Rienties, B., & Giesbers, B. (2015, June). In search for the most informative data for feedback generation: Learning analytics in a data-rich context. Computers in Human Behavior, 47, 157–167. Retrieved from http://www.sciencedirect.com/science/article/pii/S0747563214003240 doi: 10.1016/j.chb.2014.05.038

tesseract-ocr. (n.d.). Retrieved 2018-01-12, from https://github.com/tesseract-ocr

Thompson, E. (2005, February). MD5 collisions and the impact on computer forensics. Digital Investigation, 2(1), 36–40. Retrieved from http://www.sciencedirect.com/science/article/pii/S1742287605000058 doi: 10.1016/j.diin.2005.01.004

userfeel.com. (2020). Remote Usability Testing tool for Websites and Apps | Userfeel.com. Retrieved 2020-04-01, from https://www.userfeel.com/

Valderrama Bahamóndez, E. d. C., Pfleging, B., Henze, N., & Schmidt, A. (2014, September). A long-term field study on the adoption of smartphones by children in panama. In (pp. 163–172). ACM. Retrieved 2019-10-09, from http://dl.acm.org/citation.cfm?id=2628363.2628403 doi: 10.1145/2628363.2628403

van Berkel, N., Luo, C., Anagnostopoulos, T., Ferreira, D., Goncalves, J., Hosio, S., & Kostakos, V. (2016). A Systematic Assessment of Smartphone Usage Gaps. In Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (pp. 4711–4721). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2858036.2858348 doi: 10.1145/2858036.2858348

Vermeeren, A. P. O. S., Law, E. L.-C., Roto, V., Obrist, M., Hoonhout, J., & Väänänen-Vainio-Mattila, K. (2010). User Experience Evaluation Methods: Current State and Development Needs. In Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries (pp. 521–530). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/1868914.1868973 doi: 10.1145/1868914.1868973

Vuong, T., Jacucci, G., & Ruotsalo, T. (2017, November). Watching inside the Screen: Digital Activity Monitoring for Task Recognition and Proactive Information Retrieval. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, 1(3), 109. Retrieved 2019-10-09, from http://dl.acm.org/citation.cfm?id=3139486.3130974 doi: 10.1145/3130974

Wise, A. F., Speer, J., Marbouti, F., & Hsiao, Y.-T. (2013, March). Broadening the notion of participation in online discussions: examining patterns in learners' online listening behaviors. Instructional Science, 41(2), 323–343. Retrieved 2019-09-23, from https://doi.org/10.1007/s11251-012-9230-9 doi: 10.1007/s11251-012-9230-9

Yeykelis, L., Cummings, J. J., & Reeves, B. (2018, July). The Fragmentation of Work, Entertainment, E-Mail, and News on a Personal Computer: Motivational Predictors of Switching Between Media Content. Media Psychology, 21(3), 377–402. Retrieved from https://doi.org/10.1080/15213269.2017.1406805 doi: 10.1080/15213269.2017.1406805

You, J. W. (2016, April). Identifying significant indicators using LMS data to predict course achievement in online learning. The Internet and Higher Education, 29, 23–30. Retrieved 2019-08-05, from http://www.sciencedirect.com/science/article/pii/S1096751615300063 doi: 10.1016/j.iheduc.2015.11.003

Yuan, D., Zheng, J., Park, S., Zhou, Y., & Savage, S. (2011). Improving Software Diagnosability via Log Enhancement. In Proceedings of the Sixteenth International Conference on Architectural Support for Programming Languages and Operating Systems (pp. 3–14). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/1950365.1950369 doi: 10.1145/1950365.1950369

Zacharis, N. Z. (2015, October). A multivariate approach to predicting student outcomes in web-enabled blended learning courses. The Internet and Higher Education, 27, 44–53. Retrieved 2019-07-22, from https://linkinghub.elsevier.com/retrieve/pii/S1096751615000391 doi: 10.1016/j.iheduc.2015.05.002

Zauner, C. (2010). Implementation and benchmarking of perceptual image hash functions.

Öztürk, A., Bonfert-Taylor, P., & Fügenschuh, A. (2018). Using data to improve programming instruction. Gesellschaft für Informatik e.V. Retrieved 2019-03-06, from http://dl.gi.de/handle/20.500.12116/16963