Falling Asleep with Angry Birds, Facebook and Kindle – A Large Scale Study on Mobile Application Usage

Matthias Böhmer
DFKI GmbH
Saarbrücken, Germany
matthias.boehmer@dfki.de

Brent Hecht
Northwestern University
Evanston, IL, USA
brent@u.northwestern.edu

Johannes Schöning
DFKI GmbH
Saarbrücken, Germany
johannes.schoening@dfki.de

Antonio Krüger
DFKI GmbH
Saarbrücken, Germany
krueger@dfki.de

Gernot Bauer
Fachhochschule Münster
Münster, Germany
gbauer@fh-muenster.de

ABSTRACT
While applications for mobile devices have become extremely important in the last few years, little public information exists on mobile application usage behavior. We describe a large-scale deployment-based research study that logged detailed application usage information from over 4,100 users of Android-powered mobile devices. We present two types of results from analyzing this data: basic descriptive statistics and contextual descriptive statistics. In the case of the former, we find that the average session with an application lasts less than a minute, even though users spend almost an hour a day using their phones. Our contextual findings include those related to time of day and location. For instance, we show that news applications are most popular in the morning and games are at night, but communication applications dominate through most of the day. We also find that despite the variety of apps available, communication applications are almost always the first used upon a device's waking from sleep. In addition, we discuss the notion of a virtual application sensor, which we used to collect the data.

Author Keywords
Mobile apps, usage sensor, measuring, large-scale study.

ACM Classification Keywords
H.5.2 User Interfaces: Evaluation/methodology

General Terms
Human Factors, Measurement

INTRODUCTION
Mobile phones have evolved from single-purpose communication devices into dynamic tools that support their users in a wide variety of tasks, e.g. playing games, listening to music, sightseeing, and navigating. In this way, the mobile phone has become increasingly analogous to a "Swiss Army Knife" [15, 17] in that mobile phones provide a plethora of readily-accessible tools for everyday life. The number of available applications for mobile phones – so called "apps" – is steadily increasing. Today, there are more than 370,000 apps available for the Android platform and 425,000 for Apple's iPhone[1]. The iPhone platform has seen more than 10 billion app downloads[2].

Despite these large numbers, there is little public research available on application usage behavior. Very basic questions remain unanswered. For instance, how long does each interaction with an app last? Does this vary by application category? If so, which categories inspire the longest interactions with their users? The data on context's effect on application usage is equally sparse, leading to additional interesting questions. How does the user's context – e.g. location and time of day – affect her app choices? What type of app is opened first? Does the opening of one application predict the opening of another? In this paper, we provide data from a large-scale study that begins to answer these basic app usage questions, as well as those related to contextual usage.

In addition to the descriptive results above, an additional contribution of this paper is our method of data collection. All of the data for this paper was gathered by *AppSensor*, our "virtual sensor", that is part of a large-scale deployment of an implicit feedback-based mobile app recommender system called *appazaar* [4]. *appazaar* is designed to tackle the problem presented by the fact that, as mentioned above, an enormous number of apps are available. Based on the user's current and past locations and app usage, the system recommends apps that might be of interest to the user. Within the *appazaar* app we deployed *AppSensor*, that does the job vital to this research of measuring which apps are used in which contexts.

In the next section, we describe work related to this paper. Section three provides an overview of *AppSensor* and other

MobileHCI 2011, Aug 30–Sept 2, 2011, Stockholm, Sweden.

[1] Wikipedia: List of digital distribution platforms for mobile devices, http://tiny.cc/j0irz

[2] http://www.apple.com/itunes/10-billion-app-countdown/

aspects of our data collection process. In section four, we present our basic and context-related findings. Discussion of implications for design, as well as the limitations of our study, is the topic of section five. Finally, we conclude by highlighting major findings and describing future work.

RELATED WORK

Work related to this paper includes that on mobile user needs and mobile device usage and deployments in the wild. For instance, Church and Smyth [6] analyzed mobile user needs and concluded that context – in form of location and time – is important for mobile web search. Cui and Roto [7] investigated how people use the mobile web. They found that the timeframe of web sessions is rather short in general but browser use is longer if users are connected to a WLAN. Verkasalo [18] showed that people use certain types of mobile services in certain contexts. For example, they mostly use browsers and multimedia services when they are on the move but play more games while they are at home.

Froehlich et al. [10] presented a system that collects real usage data on mobile phones by keeping track of more than 140 types of events. They provide a method for mobile experience sampling and describe a system for gathering in-situ data on a user's device. The goal of Demieux and Losguin [8] was to collect objective data on the usage and interactions with mobile phones to incorporate the findings into the design process. Their framework is capable of tracking the high-level functionality of phones, e.g. calling, playing games, and downloading external programs. However, both of these studies were very limited in number of users (maximum of 16), length of study (maximum 28 days), and number of apps.

Similar to this work, McMillan et al. [16] and Henze et al. [12] make use of app stores for conducting deployment-based research. McMillan et al. [16] describe how they gather feedback and quantitative data to design and improve a game called *Yoshi*. Their idea is to inform the design of the application itself based on a large amount of feedback from end-users. Henze et al. [12] designed a map-based application to analyze the visualization of off-screen objects. Their study is also designed as a game with tasks to be solved by the players. The players' performances within different tasks are used to evaluate different approaches for map visualizations. However, app-store-based research is so far limited to single applications and has a strong focus on research questions that are specific to the deployed apps itself. In this work, we focus on gaining insights into general app usage by releasing an explorative app to the Android app store.

Another similar approach to this work is followed by the *AppAware* project [11]. The system shows end-users "which apps are hot" by aggregating world-wide occurrences of app installation events. However, since *AppAware* only gathers the installation, update, and deinstallation of an application, the system is not aware of the actual usage of a specific app.

In summary, this research is unique (to our knowledge) in that it combines the approach of large-scale, in-the-wild user studies with the fine-grained measuring of app usage. In this way, we are able to (1) study large numbers of users and (2) large numbers of applications, all over a long time period. Previous work has had to make sacrifices in at least one of these dimensions, as Table 1 shows. Furthermore, the mobile phones used in related studies have been mostly from the last generation, i.e. they could not be customized by the end-users in terms of installing new applications.

APPSENSOR AND DATA COLLECTION

In this section, we describe our data collection tool, *AppSensor*. Because context is a known important predictor of the utility of an application [3], *AppSensor* has been designed from the ground up to provide context attached to each sample of application usage.

Lifecycle of a Mobile App

In order to understand the *AppSensor*'s design, it is important to first give the *AppSensor*'s definition of the lifecycle of a mobile application (Figure 1). The *AppSensor* understands five events in this lifecycle: installing, updating, uninstalling, opening, and closing the app.

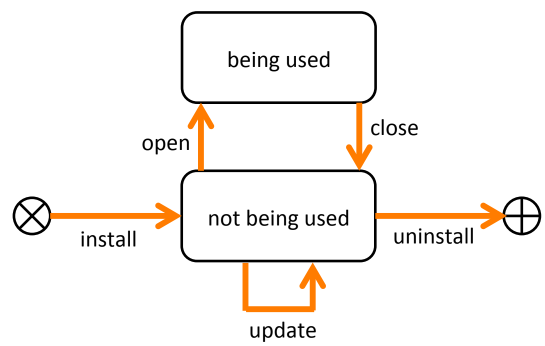

Figure 1. The lifecycle of a mobile app on a user's device according to different states and events.

The first event that we can observe is an app's installation. It reveals that the user has downloaded an app, e.g. from an app market. Another event that is observable is the update of an app, which might be interpreted as a sign of enduring interest in the application. However, since updates are sometimes done automatically by the system and the update frequency strongly depends on the release strategy of the developer, the insight into usage behavior that can be gained from update events is relatively low. The last event we can capture is the uninstall event, which expresses the opposite of the installation event: a user does not want the app anymore.

However, these maintenance events only occur a few times per app. For some apps, there might even be only a single installation event (e.g. when the user has found a good app) or even none at all (e.g. for preinstalled apps like the phone app). Maintenance events are also of limited utility for understanding the relationship between context and app usage. For instance, a user might install an app somewhere but use it elsewhere (e.g. an app for sightseeing that is installed at home before traveling).

	Users	Apps	Days	Comment
Verkasalo [18]	324	$\sim$14	67	Investigation of contextual pattern of mobile device usage.
Froehlich et al. [10]	4-16	-	7-28	System for collecting in-situ data (pre-installed).
Demieux and Losguin [8]	11	-	2	A study with a strong focus on device usage (distributed via SMS).
Girardello & Michahellis [11]	19,000	-	-	Measuring popularity instead of usage (released to Android Market).
McMillan et al. [16]	8,674	1	154	Exploring world-wide field trials (released to iPhone App Store).
Henze et al. [12]	3,934	1	72	Evaluation of off-screen visualization (released to Android Market).
AppSensor (this paper)	4,125	22,626	127	Large scale study on app usage (released to Android Market).

Table 1. Overview of related app-based studies conducted in-situ on participants' devices. The table shows fine grained usage analysis (rows 1-3) and large-scale studies (rows 4-6).

Instead, *AppSensor* is designed to continuously sample a user's application *usage*. In other words, we are especially interested in the two app states of *being used* and *not being used*, which can both be inferred from the open and close events. These events naturally appear much more often and in a much shorter period of time than the maintenance events. They enable us to observe app usage on a more fine-grained level and provide a much more accurate understanding of context's effects on app usage.

In order to gather data on the *being used* and *not being used* states, *AppSensor* takes advantage of the fact that the Android operating system can report the most recently started application. Because of this feature, we know the app with which the user is currently interacting. We are thus able to infer which single app is in the state of *being used* owing to the fact that the Android operating system only shows one app to the user (as does the iPhone OS). Therefore, we can presume that all other applications are consequently in the state of *not being used* in terms of not showing their graphical interface. In this study, we do not consider background applications that are not interacted with through a graphical user interface, e.g. background music apps that can be controlled through gestures.

Formal Description of AppSensor

As noted above, the *AppSensor* is meant to be a sensor that indicates the currently used application at a given time t. Formally speaking, the sensor can be described as follows: Let $A = \{a_1, \ldots, a_n\}$ be the set of apps that are available for a mobile device and let $A^* = A \cup \{\epsilon\}$ be the set of applications with which a user can interact. ϵ means that the user is currently not using any application. For most current platforms, e.g. Google's Android, this set is usually defined by the applications available on the corresponding application stores. Since the number of applications is growing, this set is not static, but has a defined number n of elements. With time given as t, the *AppSensor* shall provide the following values:

$$as(t) = \begin{cases} a_i & \text{if app } a_i \text{ is used,} \\ \epsilon & \text{if no app is used.} \end{cases}$$

With respect to the lifecycle of mobile apps the value $as(t)$ describes the application with which a user is currently interacting. The value is distributed on the nominal scale given by the set A^* of available applications. Therefore, the only conclusion that can be drawn on the mere sensor data of two measures at times t_1 and t_2 is a comparison on whether the application a user is running is the same as before (if $as(t_1) = as(t_2)$) or whether it has changed (if $as(t_1) \neq as(t_2)$).

Implementation and Deployment

AppSensor is implemented as a background service within Android and is installed by end users as part of the *appazaar* application. This app traces context information that is available directly on the user's device (e.g. location, local time, previous app interaction) and app usage at the same time. The recommender algorithms of appazaar rely on this data and *appazaar's* app was the means for enabling the data collection reported in this paper. The applied sampling rate is 2 Hz. *AppSensor* collects data every $500ms$ in a loop that starts automatically as soon as the device's screen is turned on and stops when the screen is turned off again. When the device goes into standby-mode[3], we consider which app was left open and omit the standby time from the application's usage time. The measured data is written to a local database on the user's device and only periodically uploaded to a server. In case of connectivity failure, the data is kept in the database and attached to the next transmission.

The first version of *appazaar* was released to the Android Market in May 2010. In August 2010, we released a version with the *AppSensor* as presented in this paper. Of course, the data collected by *AppSensor* is primarily designed to provide "the best app recommendation" within the *appazaar* application, i.e. to inform the recommendation process of apps to a user in a given context [5]. For security and privacy reasons, the system uses hash functions to anonymize all personal identifiers before the data is collected, and we do not query any additional personal information like the name, age or sex from the user.

Application Categorization

In order to get a more high level understanding of our data, we felt it was necessary to add categories to the applications opened by our users. To do so, we mined the Android Market for each app's category (see Table 2). As such, the categories are largely given by the apps' developers: they – as domain experts – assign their apps to the categories when uploading them to the Android market. The only exception to this rule occurred in some minor manual modifications. For instance, we merged all games of the categories *Arcade & Action*, *Brain & Puzzle*, *Cards & Casino*, and *Comics* into

[3]Determined by screen-off and screen-on events.

one *Games* category. Due to the special nature of browsers – they do not have clear cut domain scope – we have separated them into their own dedicated *Browsers* category. For some apps, no categories are available on the Android Market. These are either test applications by developers that appear only on a few devices, applications that are distributed via other channels (e.g. pre-installed by device manufacturers), default Android apps (e.g. settings), or apps that have been taken out of the market and whose category was not available anymore[4]. We manually added categories for some apps where possible. For the branded Twitter clients of some vendors (e.g. HTC), we added the category of the original Twitter app (*Social*). To the default apps responsible for handling phone calls we added the *Communication* category. As we did with the browser, we also put the settings app into its own category (*Settings*) due to its special nature. Since the main menu on Android phones itself is also an app and it is treated as such from the system's perspective, we additionally removed such launcher apps from the results since they give little insight into app usage behavior. Finally, it is important to note that each app can only have one category.

Characteristics of Final Dataset

The results reported in this paper are based on data from the 4,125 users, who used *appazaar* between August 16th, 2010 and January 25th, 2011. The users were spread out geographically, although most stayed in the United States or Europe during our study (see Figure 2). Within the timeframe of 163 days, they generated usage events for 22,626 different applications and the deployment of our *AppSensor* was able to measure 4.92 million values for application usage. We advertised *appazaar* on Facebook and Twitter and two posts about the system appeared on two well-known technology blogs (*Gizmodo* and *ReadWriteWeb*), helping us reach a growing number of users.

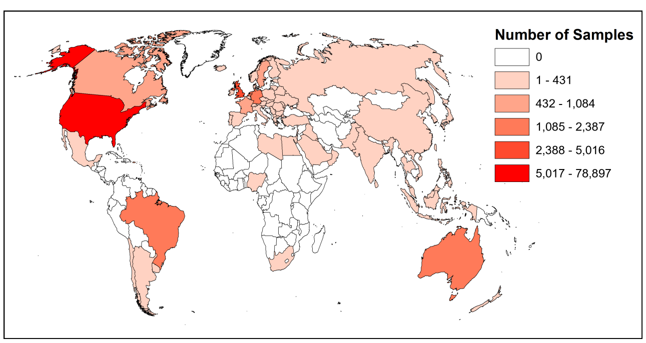

Figure 2. The geographic distribution of our users. Data classes determined via ESRI ArcMap's 'natural breaks' algorithm, a well-known standard in cartography and geovisualization that is helpful in accurately displaying the underlying distribution of the data.

RESULTS

This section is divided into two parts: (1) basic descriptive statistics on application usage behavior and (2) context-sensitive statistics. In the second section, we look at several different forms of context, including an application's place in an "app use chain", as well as more standard contextual variables such as time and location. In both sections, our primary resolution of analysis is the "application category" as defined above, but in the second section we do highlight some interesting application-level temporal patterns.

Basic Descriptive Statistics

On average, our users spent 59.23 minutes per day on their devices. However, the average application session – from opening an app to closing it – lasted only 71.56 seconds.

Table 2 shows the average usage time of apps by category, which ranged from 36.47 seconds for apps of *unknown* category and 31.91 seconds for apps of category *Finance* to 274.23 seconds for category *Libraries & Demos*. The most-used *Libraries & Demos* apps as measured by total usage time are inherent apps of the operating system (*Google Services Framework*, *default Updater*, *Motorola Updater*). It was interesting to see that this category has a much longer average session than the games category, whose most used applications are *Angry Birds*, *Wordfeud FREE*[5], and *Solitaire*. On the low end of the session length spectrum of apps with known categories, we found the *Finance* category. The most used apps of this category are for personal money management (*Mint.com Personal Finance*), stock market (*Google Finance app*), and mobile banking (*Bank of America*). The briefness of the average session in this category does not speak well for the success rate of financial applications on the Android platform.

Category	Apps	Avg. usage	Examplary Apps
unknown	4,823	36.37 sec	-
Finance	307	37.01 sec	Mint.com Personal Finance, Bank of America, Google Finance, iStockManager
Travel	782	44.72 sec	Google Maps, Yelp, Waze
Communication	881	46.92 sec	Google Mail, Handcent SMS, K-9 Mail
Productivity	1,062	61.49 sec	Calendar, Evernote, GTasks
Shopping	326	61.71 sec	Market, Barcode Scanner, Craigslist
Social	538	62.69 sec	Facebook for Android, Twitter, TweetDeck
Sports	385	65.98 sec	Yahoo! Fantasy Football '10, ESPN ScoreCenter, NFL Mobile
News	784	68.11 sec	NewsRob, reddit is fun, BBC News
Settings	1	68.71 sec	Default Settings App
Browser	10	74.01 sec	Default Browser, Skyfire Browser, Dolphin Browser
Entertainment	84	76.90 sec	IMDb Movies & TV, TV Guide Mobile, PhotoFunia
Multimedia	130	82.79 sec	Pandora Radio, Music, Camera
Comics	3,242	91.33 sec	DailyStrip, XkcdViewer, Dilbert Mobile
Games	2,822	114.25 sec	Angry Birds, Wordfeud FREE, Solitaire
Health	424	153.80 sec	CardioTrainer, Sleep Bot Tracker Log, Baby ESP
Lifestyle	956	167.77 sec	DailyHoroscope, Gentle Alarm, Epicurious Recipe
Reference	764	176.28 sec	Kindle for Android, Aldiko Book Reader, Audible
Tools	3,004	206.26 sec	AppBrain App Market, Apps Organizer, Google Goggles
Themes	1,061	258.28 sec	Zune Home, Fingerprint Screensaver, HomeChange
Libraries & Demos	240	274.23 sec	Google Services Framework, default Updater, Motorola Updater, Bubbles Demo, Ride Logger Demo, ES Task Manager

Table 2. Number of apps investigated in our study and average usage time of every categories' apps from opening to closing.

[4]We crawled the Android Market on February 3rd, 2011.

[5]Despite its name *Wordfeud FREE* is a full game and not a demo version since it provides the same full functionality like the non-free version. The only difference is that it is for free and contains advertisements.

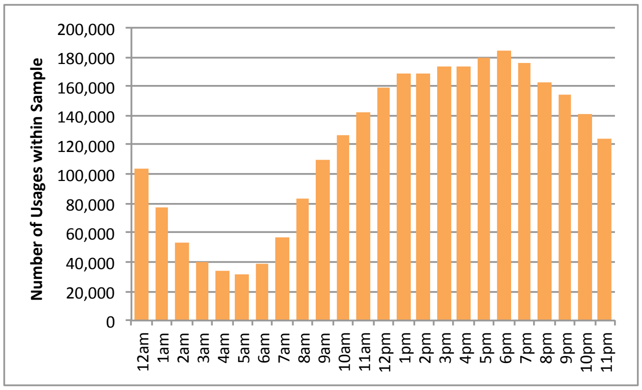

Figure 3. Total number of recorded app utilizations during a day.

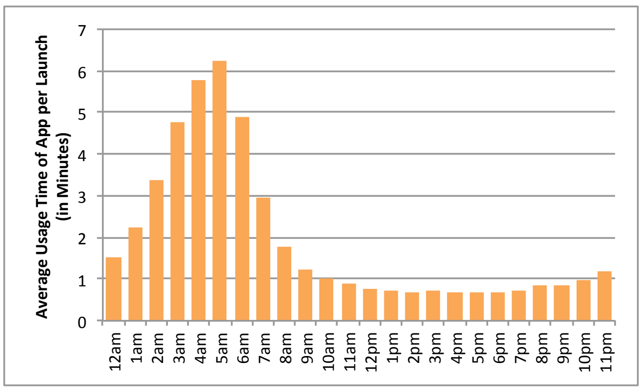

Figure 4. Daily average usage duration of opened apps per launch in minutes.

Contextual Results: Application Usage over Time

AppSensor allows us to record temporal information about application usage. Figure 3 shows the total number of application launches in our sample according to hour of the day. It can be seen that total application usage (in terms of launches) is at its maximum in the afternoon and evening, peaking around 6pm. Our participants generally start using applications in the morning between 6am and 7am, and their activity grows approximately linearly until 1pm. Activity increases slowly to a peak around at 6pm. Application usage minimum is around 5am, although it never falls below 16% of its maximum.

Figure 4 shows the average time people spent with an application once they opened it with regard to the hour of the day. There is a peak around 5am with 6.26 minutes of average app usage time. The average application session is less than a minute, however, reaching a minimum of around 40 seconds at 5pm. Interestingly, the graph in Figure 4 is nearly opposite that in Figure 3. This means, that when people actively start to use their devices, they spend less time with each application. This might be due to apps that people explicitly leave active while sleeping with standby-mode prevented, but there are other possible explanations.

Figure 5 shows the change in the relative usage of the application categories over the course of the day in terms of number of app launches. Mobile devices are most likely to be used for communication every hour of the day, especially in the afternoon and evening (11am-10pm) with a probability of more than 50%. News apps have the highest probability of being used in the morning (from 7am to 9am). Around 11am, finance apps briefly become quite prominent. After communication winds down late in the evening, games have their highest probability of use. Social applications also have their highest probability of use in the late evening (from 9pm to 1am). Sports apps seem to play their most important role in the afternoon (2pm-5pm) and evening (8pm-10pm). During the early morning, when total application usage is at its lowest, people share their time with apps of various categories. This is also the time when communication app use share is minimized.

Contextual Results: Chains of App Usage

An important contextual variable in usage behavior are the zero or more applications used before an application is opened and the zero or more applications used afterwards. We defined an "application chain" as a sequence of apps that are used without the device being in standby mode for longer than 30 seconds. In total, we can distinguish 1,841,226 such sessions in our data set. Examples include one in which a user started with *Grocery iQ (Shopping)*, switched to *GrubHub Food Delivery (Lifestyle)*, and ended with *Epicurious Recipe App (Lifestyle)*. Another user started with the *AroundMe (Lifestyle)* app and then continued with *Find A Starbucks (Shopping)*, *Google Maps (Travel)*, *Find A Starbucks*, *Google Maps*, *Find A Starbucks*, *Dolphin Browser HD (Browser)*, *Find A Starbucks*, *Google Maps*, *Find A Starbucks*, and *Google Maps*.

Figure 6 demonstrates the distribution of application chains by the number of applications that occur in the chain. As the y-axis of Figure 6 is on a log-scale, it can be seen that the majority of sessions (68.2%) only contain a single application. In other words, people turn on their phone, use a single app, and put their phone back in standby. This tendency towards the use of a small number of applications during an interaction with the mobile device is further evidenced by the fact that only 19.5% of application chains contain two apps, and only 6.6% contain three.

We also looked into the number of *unique* apps used within a session, as can be seen in Figure 7. The first bar in this figure is of course identical to the first bar in the preceding figure. We found a maximum of 14 unique apps in an app chain. A vast majority of users use a very small number of unique apps during an interaction with their device. Thus – according to our analysis of sessions – people who use more than 14 apps in sequence tend to re-use apps they already have used before within the same session.

Examining the amount of time our users spent in each application chain, we found that 49.8% of all recorded sessions are shorter than 5 seconds. The longest session we observed has a length of 59 minutes and 48 minutes. Between these two end points, the curve has a similar exponential decay to that in Figure 6 and 7.

	12am	1am	2am	3am	4am	5am	6am	7am	8am	9am	10am	11am	12pm	1pm	2pm	3pm	4pm	5pm	6pm	7pm	8pm	9pm	10pm	11pm	% of Total Launches	Users	Apps
Browser	7.9%	7.7%	7.8%	7.6%	7.3%	7.4%	7.0%	7.9%	8.1%	8.0%	7.7%	7.3%	7.0%	6.9%	6.8%	6.4%	6.6%	6.6%	6.4%	6.6%	7.0%	7.4%	7.5%	7.4%	6.83%	2,398	9
Comics	4.5%	5.2%	5.4%	5.8%	5.8%	5.6%	5.5%	5.2%	5.4%	5.1%	4.7%	4.3%	4.3%	4.2%	4.2%	4.3%	4.4%	4.0%	4.4%	4.2%	4.1%	4.1%	4.1%	4.4%	4.31%	2,151	1,810
Communication	44.9%	41.1%	38.3%	35.4%	31.6%	31.8%	32.7%	34.7%	39.4%	44.8%	49.0%	52.6%	54.8%	55.2%	55.2%	56.1%	55.7%	56.8%	57.1%	56.1%	54.8%	53.3%	52.0%	49.0%	49.50%	2,769	550
Entertainment	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.02%	126	43
Finance	0.2%	0.3%	0.3%	0.2%	0.1%	0.1%	0.1%	0.2%	0.3%	0.3%	0.4%	0.5%	0.3%	0.3%	0.4%	0.3%	0.3%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.25%	604	164
Games	3.2%	3.0%	3.0%	2.7%	2.5%	2.3%	2.2%	1.7%	1.9%	1.9%	2.0%	2.1%	2.2%	2.2%	2.2%	2.3%	2.3%	2.2%	2.2%	2.4%	2.7%	3.0%	3.0%	3.2%	2.30%	1,716	1,702
Health	0.3%	0.4%	0.4%	0.4%	0.6%	0.6%	0.7%	0.6%	0.4%	0.3%	0.3%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.3%	0.2%	0.3%	0.2%	0.3%	0.26%	540	227
Libraries & Demo	0.4%	0.5%	0.6%	0.7%	0.9%	0.8%	0.7%	0.6%	0.5%	0.4%	0.3%	0.3%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.2%	0.3%	0.3%	0.3%	0.3%	0.30%	1,267	117
Lifestyle	0.8%	0.9%	1.0%	1.4%	1.3%	1.5%	1.4%	1.4%	1.1%	0.9%	0.6%	0.6%	0.5%	0.5%	0.5%	0.5%	0.6%	0.5%	0.3%	0.4%	0.4%	0.5%	0.5%	0.5%	0.60%	2,132	451
Multimedia	2.1%	2.1%	2.4%	2.4%	2.7%	2.4%	1.8%	1.8%	1.9%	1.7%	1.8%	2.0%	2.0%	2.0%	2.2%	2.1%	2.2%	2.4%	2.3%	2.3%	2.2%	2.1%	1.9%	2.0%	2.03%	1,713	76
News	2.6%	2.5%	2.6%	2.5%	2.5%	2.7%	3.3%	3.7%	4.1%	3.6%	3.0%	2.6%	2.5%	2.7%	2.5%	2.4%	2.2%	2.1%	2.3%	2.2%	2.3%	2.2%	2.3%	2.3%	2.46%	1,777	440
Productivity	3.6%	5.0%	5.0%	5.8%	6.3%	6.5%	6.0%	5.4%	4.8%	5.1%	4.9%	4.3%	4.2%	4.0%	4.0%	3.7%	3.4%	3.4%	3.0%	3.1%	3.1%	3.0%	2.9%	3.2%	3.76%	2,190	648
Reference	0.7%	0.7%	0.7%	0.7%	0.7%	0.7%	0.6%	0.6%	0.7%	0.5%	0.5%	0.5%	0.4%	0.4%	0.4%	0.4%	0.3%	0.4%	0.4%	0.4%	0.5%	0.5%	0.5%	0.6%	0.47%	903	346
Settings	1.3%	1.6%	1.5%	1.3%	1.6%	1.2%	1.2%	1.1%	1.3%	1.4%	1.4%	1.4%	1.2%	1.3%	1.2%	1.2%	1.3%	1.1%	1.1%	1.2%	1.2%	1.3%	1.3%	1.4%	1.23%	2,178	1
Shopping	3.9%	4.5%	3.7%	3.4%	3.2%	3.2%	3.1%	3.0%	3.1%	3.3%	3.2%	3.2%	3.2%	2.8%	2.9%	2.9%	2.7%	2.7%	2.7%	2.7%	2.8%	3.1%	3.6%	3.5%	2.96%	2,556	198
Social	5.7%	5.0%	4.9%	4.3%	4.2%	4.0%	4.4%	5.1%	5.3%	5.4%	5.2%	5.0%	4.7%	4.8%	4.9%	4.5%	4.5%	4.6%	4.6%	4.9%	5.2%	5.4%	5.8%	5.7%	4.77%	1,902	342
Sports	0.5%	0.3%	0.3%	0.2%	0.3%	0.3%	0.2%	0.3%	0.3%	0.3%	0.3%	0.4%	0.4%	0.6%	0.7%	0.8%	0.9%	0.8%	0.6%	0.6%	0.7%	0.8%	0.7%	0.7%	0.56%	571	215
Themes	0.2%	0.1%	0.2%	0.3%	0.4%	0.4%	0.4%	0.2%	0.2%	0.2%	0.1%	0.1%	0.1%	0.2%	0.1%	0.1%	0.1%	0.2%	0.1%	0.1%	0.2%	0.1%	0.1%	0.1%	0.14%	249	231
Tools	10.9%	12.2%	14.6%	17.6%	20.3%	21.5%	21.4%	18.6%	14.7%	10.4%	8.4%	6.8%	6.1%	5.9%	5.9%	5.9%	6.0%	6.1%	5.8%	6.0%	6.3%	6.8%	7.4%	9.1%	7.89%	2,512	1,688
Travel	1.4%	1.6%	2.1%	2.2%	2.4%	2.6%	2.2%	1.9%	2.0%	2.1%	2.0%	1.8%	1.9%	1.9%	1.9%	1.8%	2.0%	1.9%	2.2%	2.2%	1.9%	1.7%	1.6%	1.4%	1.86%	1,752	407
Unknown	4.7%	5.3%	5.1%	5.0%	5.3%	4.4%	5.0%	5.9%	4.6%	4.4%	4.1%	3.8%	3.5%	3.8%	3.7%	3.7%	4.0%	3.6%	3.7%	3.7%	3.7%	3.9%	4.1%	4.5%	3.88%	2,284	1,796
Total Launches per Hour	103,604	77,053	53,633	40,332	33,438	30,949	38,161	56,895	83,488	109,550	127,069	142,642	158,876	168,082	169,018	172,935	173,963	179,801	184,012	176,050	163,080	153,835	141,303	123,639			

Figure 5. Hourly relative app usage by category in terms of launches. Each cell value refers to the percentage of app launches done by our users within each hour for each category. Colors are normalized by row, with green indicating each category's maximum percentage of application time, and white indicating each category's minimum. For example, games reach their peak in the evening (green) and trough in the morning (white).

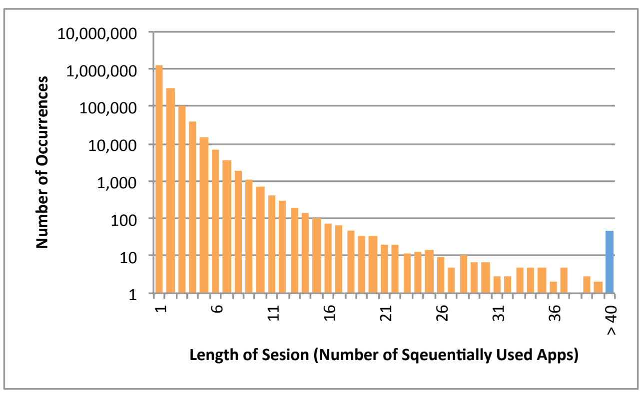

Figure 6. Number of apps used in a session. We aggregated sessions longer than 40 apps since the graph flattens out and scarcity increases. Maximum length is 237.

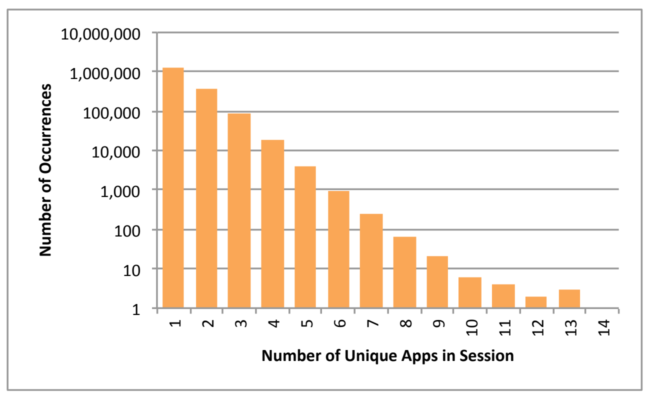

Figure 7. Occurrences of sessions according to number of unique apps used within a session.

Probably the most revealing statistic in our analysis of application chains is that for nearly half of all chains (49.60%) the first application belongs to the category *Communication* (as Figure 8 shows). Digging deeper, we found that 15.7% of the chains within our sample were initiated with an SMS application (9.5% default sms app, 6.2% an app called *Handcent SMS*), 9.6% with the phone application, and 5.9% with the standard mail application. Interestingly, a browser was only used first in 5.9% of the application chains.

Figure 9 shows the transition probabilities between application categories in an application chain. Accordingly, the diagonal of the figure indicates transitions from one app to another in the same category. As such, the values along the diagonal are non-zero. This graph considers only those sessions where two or more apps have been used. For each app, it is very likely that the app used next is a communication

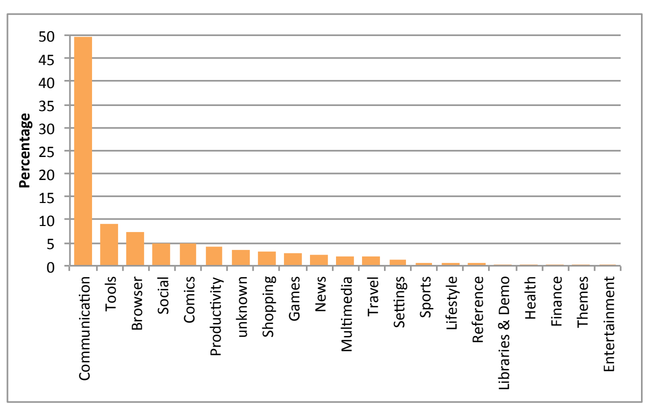

Figure 8. Categories of first used app within a session.

app, except for *News* and *Lifestyle* applications. Apart from these two categories, the probability that the next app is a communication app is at least 23.2% for all categories. For communication apps, there is a 66.5% chance that the next app will be a communication app again. This is the highest probability for users to stay within one category. Next are *Tools*, with a probability of 15.7% of staying within *Tools*, and *Games* with a probability of 15.1% of staying within *Games*. It can also be seen that apps from category *Tools* are entered relatively frequently from apps of any category.

There are also some important unique connections between application categories. Most notably, a browser is opened quite frequently following the use of a news application. The connection between *Lifestyle* and *Shopping* applications is also quite strong, with *Lifestyle* applications frequently leading the user to enter into a *Shopping* application. The reverse is also true, but to a lesser extent.

Contextual Results: Application Usage by Location

We also found clear empirical evidence for location as a covariate of app usage behavior. This occurs across changes in both administrative regions (e.g. USA vs. Europe) and functional regions (e.g. airports vs. outside of airports). We present some initial findings from our spatial analysis below.

We examined 13,190 samples recorded by the *AppSensor* that occurred while a user was located within the spatial footprint of a known airport in the United States. We found that while in the airport, users were 2.78 times more likely to be using a browser (by usage time) than a user located in the United States as a whole. This may suggest that certain functions related to air travel may not be sufficiently migrated into native applications (e.g. looking up flight status). On the other hand, users were less likely to be using games, tool applications, or reference applications while in airports. This was somewhat surprising, especially given that the *Kindle* app belongs to the reference category.

When traveling at speeds greater than 25kph, we found, not surprisingly, that users were more than 2.26 more likely (by usage time) to be using an app of the *Multimedia* category, to which most music-related applications belong. Interestingly, we found that they were less likely (0.83) to be using apps in the *Travel* category.

We found some interesting differences between users in the United States and in Europe. European users are 1.21 times more likely to be found using a browser (by usage time). Americans, however, spent relatively much more time with sports, health, and reference applications. Social and news apps were the most equally used.

Specific Application Usage

Although we focused our analysis at the application category level, we did analyze several important and/or well-known individual applications. Figure 10 shows the usage times of specific applications with regard to time. In contrast to Figure 5, the numbers in Figure 10 are not normalized by total usage over all apps within the hours, but by each app's total usage per day.

Previously, we saw that social apps in general have their highest probability to be used in the evening. This is somewhat true for Facebook, but its usage time is spread out throughout the whole day. The same goes for Twitter, although it is not as much of a late-night activity.

A somewhat surprising finding can be found in the usage of the *Google Maps* app (*Travel*), which has a relatively strong peak in the early evening hours. Traffic checking is perhaps one possible cause, although one would expect this pattern to be repeated during the morning commute. Another interesting result comes from the built-in Music app's use, which is somewhat focused in the morning hours.

Weather checking is, not surprisingly, largely a morning activity, as is the checking of one's calendar. On the other hand, users' desire to fling *Angry Birds*[6] at pigs is absent in the morning, and only picks up in the early afternoon and into the evening. *Kindle* usage behavior is even more focused in the late evening.

Another interesting phenomenon emerged from the study of two different alarm clock apps. It seems, that alarm clock apps are mostly used – i.e. being the only active app on the device presenting its user interface – during the night (from 2am until 9am). One reason for this might be, that people "use" the app while sleeping, e.g. as a desk clock preventing the device from going into standby mode.

More generally speaking, Figure 10 shows that some apps have spikes in usage, whereas others are more broadly employed throughout the day.

DISCUSSION

Implications for Design

The results reported here could be used to improve the design of mobile applications and mobile operating systems. For instance, designers of "launcher" apps (like the home screen on the iPhone and Android) could vary app icon position and size based on time of day and/or location. This same idea could apply with regard to an application chain with the last app opened providing the context rather than time/location. Similarly, app developers could design smart links between apps that are used frequently in sequence. Since people often navigate from lifestyle apps to shopping apps, the designers of the former might implement links to shopping apps. Additionally, the *AppSensor* gives insights into the apps' contexts of use. For instance, the design of apps can be optimized if the developers know whether an app is used only while commuting or solely in the evening.

Our results show that mobile phones are still first and foremost communication devices. This is not only due to phone calls, as smart phones provide a variety of new ways to communicate (e.g. instant messengers, email, voice over IP). Nevertheless, this finding certainly qualifies the mobile

[6]See http://market.android.com/details?id=com.rovio.angrybirds

	Browser	Comics	Communication	Entertainment	Finance	Games	Health	Libraries & Demo	Lifestyle	Multimedia	News	Productivity	Reference	Settings	Shopping	Social	Sports	Themes	Tools	Travel	Unknown	Samples	Users	Apps
Browser	2.4%	3.6%	33.8%	0.0%	0.3%	3.5%	0.2%	0.2%	0.4%	1.5%	11.8%	3.8%	0.6%	1.7%	3.6%	15.6%	0.5%	0.3%	8.1%	2.2%	6.1%	48,379	2,193	9
Comics	6.5%	9.4%	36.1%	0.0%	0.2%	4.8%	0.6%	0.2%	0.6%	5.2%	2.7%	4.1%	0.6%	2.2%	5.2%	4.3%	0.6%	0.4%	8.4%	2.7%	5.0%	31,258	1,754	1,220
Communication	5.7%	2.7%	65.5%	0.0%	0.2%	1.5%	0.1%	0.1%	0.2%	1.3%	2.1%	2.5%	0.3%	1.0%	1.7%	4.8%	0.4%	0.1%	5.0%	1.4%	3.2%	434,974	2,839	449
Entertainment	6.7%	6.1%	26.1%	0.0%	0.0%	3.3%	0.6%	0.0%	0.6%	5.6%	0.6%	2.8%	0.0%	3.3%	7.2%	3.3%	3.3%	0.0%	8.3%	5.6%	16.7%	180	65	28
Finance	10.3%	3.7%	37.3%	0.0%	1.8%	2.9%	0.2%	0.3%	0.2%	1.5%	8.6%	3.5%	0.1%	1.5%	5.5%	6.1%	0.7%	0.1%	10.6%	1.9%	3.1%	1,496	347	117
Games	11.8%	5.9%	30.4%	0.0%	0.3%	15.1%	0.3%	0.4%	0.7%	1.0%	2.1%	4.2%	0.7%	1.5%	6.5%	4.0%	0.8%	0.1%	8.3%	1.7%	4.2%	8,620	1,077	995
Health	3.8%	4.8%	34.3%	0.0%	0.3%	2.5%	6.1%	0.6%	1.2%	6.1%	2.9%	3.1%	1.6%	2.3%	6.0%	4.9%	0.8%	0.0%	12.4%	2.3%	3.9%	1,466	328	130
Libraries & Demo	6.0%	3.7%	23.3%	0.0%	0.2%	2.3%	0.3%	2.6%	0.8%	1.3%	1.7%	3.2%	0.3%	16.2%	11.9%	3.7%	0.3%	0.1%	13.4%	3.2%	5.5%	3,936	1,082	90
Lifestyle	8.2%	5.3%	17.3%	0.0%	0.1%	4.0%	0.5%	0.6%	3.0%	0.9%	2.3%	4.3%	0.7%	2.3%	28.7%	3.1%	0.2%	0.4%	10.2%	2.2%	5.5%	4,673	1,383	303
Multimedia	6.2%	10.5%	38.2%	0.0%	0.2%	1.4%	0.6%	0.2%	0.4%	2.5%	2.5%	6.2%	0.3%	2.0%	1.8%	4.4%	0.3%	0.4%	9.5%	3.2%	9.1%	12,451	1,376	53
News	33.6%	3.3%	33.3%	0.0%	0.5%	1.6%	0.2%	0.1%	0.2%	1.4%	3.9%	2.9%	0.4%	1.4%	3.0%	3.7%	0.4%	0.0%	6.5%	1.0%	2.4%	25,131	1,440	312
Productivity	7.4%	5.0%	38.5%	0.0%	0.4%	2.6%	0.4%	0.2%	0.6%	2.8%	2.8%	7.2%	1.1%	3.8%	4.8%	5.1%	0.6%	0.3%	9.7%	2.4%	4.4%	31,113	1,954	498
Reference	13.1%	4.5%	34.3%	0.0%	0.2%	7.5%	0.6%	0.3%	1.0%	1.0%	2.5%	4.6%	2.9%	1.7%	5.2%	4.1%	0.4%	0.2%	9.8%	1.7%	4.4%	2,611	552	199
Settings	8.9%	5.6%	26.3%	0.1%	0.2%	1.8%	0.4%	5.2%	0.7%	2.0%	2.6%	6.9%	0.5%	0.0%	5.6%	4.7%	0.6%	0.5%	11.6%	4.8%	11.1%	13,576	1,863	1
Shopping	8.5%	7.8%	23.2%	0.0%	0.4%	4.8%	0.4%	0.9%	9.6%	0.9%	2.8%	5.2%	0.7%	3.0%	4.7%	4.3%	0.5%	0.5%	16.6%	1.6%	3.8%	21,788	2,207	132
Social	24.1%	3.0%	35.3%	0.0%	0.3%	2.3%	0.2%	0.2%	0.3%	1.2%	2.9%	2.8%	0.3%	1.5%	2.7%	12.4%	0.7%	0.1%	5.3%	1.2%	3.3%	35,086	1,593	239
Sports	7.4%	4.3%	43.3%	0.1%	0.4%	2.5%	0.4%	0.2%	0.3%	1.3%	3.0%	4.8%	0.5%	2.4%	3.8%	5.4%	7.6%	0.0%	7.0%	1.5%	3.9%	2,793	387	135
Themes	8.5%	10.2%	37.2%	0.0%	0.2%	2.4%	0.1%	0.2%	1.4%	3.2%	0.4%	4.7%	0.4%	3.3%	6.5%	3.6%	0.1%	1.2%	8.6%	3.3%	4.6%	1,929	175	175
Tools	11.0%	5.1%	36.1%	0.0%	0.2%	2.7%	0.3%	0.4%	0.6%	2.1%	2.4%	4.2%	0.6%	2.1%	5.5%	4.1%	0.4%	0.2%	15.7%	2.8%	3.5%	88,911	2,384	1,310
Travel	6.7%	9.1%	36.2%	0.1%	0.2%	2.3%	0.3%	0.5%	0.7%	1.9%	1.6%	6.7%	0.4%	5.0%	2.9%	4.4%	0.3%	0.2%	10.2%	6.6%	3.6%	12,556	1,403	281
Unknown	10.7%	4.4%	40.8%	0.1%	0.2%	2.1%	0.2%	0.3%	0.6%	3.9%	1.8%	3.2%	0.3%	3.9%	2.9%	4.7%	0.3%	0.2%	6.4%	1.5%	11.6%	48,379	1,972	1,277

Figure 9. Transition probabilities in app chains. The transitions are from categories in a row to categories in a column. The diagonal indicates transitions between apps in the same category. The probability ranges from yellow (low) to green (high).

phones as "Swiss Army Knives" line of thinking. That said, when people are not sleeping during the late hours of the night they make more use of the non-communication functionality provided by different kinds of apps. Additionally, they spend more time within an app once they have opened it in the night.

Our results also suggest that users frequently switch between already used apps in application chains rather than only opening new apps. This suggests that there is a functional cohesion between the particular utilizations of single apps. As such, mobile phone operating systems should better support navigation between very recently used apps.

Making Use of the AppSensor

The *AppSensor* gives rise to examining the eco-system of apps residing on a user's device, this has potential to inform the design and customization of novel applications as well as new devices itself.

One may apply the *AppSensor* for inferring a user's context based on his actually used apps. According to Dey [9], context-awareness involves adapting services according to a user's context. For instance, the users' needs for mobile services – i.e. apps in our case – depend on their locations [14]. We propose that by adding the *AppSensor* to context-reasoning one can decrease the uncertainty of context recognition. For instance, even though two people may be walking through the same pedestrian mall in a famous city (i.e, same location), if they use different apps (e.g. a shopping list app vs. a sightseeing app) we can distinguish between the shopper and the tourist. Even without any meta-information on the used apps itself, it would be possible to compare the contexts of two or more people. For instance, if two users are constantly swapping between a map app and a restaurant guide app they might be in the same activity – probably looking for a restaurant.

Context-aware recommender systems that suggest mobile applications can be made more efficient by exploiting an *AppSensor*. This was our main motivation for conducting the presented study. For instance, recommender systems that follow a post-filtering approach – i.e. applying knowledge on context-aware dependencies after using basic techniques like collaborative filtering [1, 13] – can exploit the time-dependent usage share as factor on the estimated ranking of apps.

Limitations

Some apps have a more general purpose that is not well understood by *AppSensor*. For instance, a web browser can be used for everything from public transportation route planning to looking up a word in a dictionary. The meaning that can be deduced from such applications can be regarded as limited or imprecise. For these cases, the insight that the *AppSensor* provides on the user's context might be limited. However, most services that are provided via a browser are also available within dedicated applications. Since many users seem to prefer to employ native apps instead of websites on mobile devices [2], this should not have a large negative impact.

	12am	1am	2am	3am	4am	5am	6am	7am	8am	9am	10am	11am	12pm	1pm	2pm	3pm	4pm	5pm	6pm	7pm	8pm	9pm	10pm	11pm	% of Total Usage Time	Users
Facebook	4.8%	3.9%	4.0%	3.4%	3.2%	3.3%	3.8%	4.1%	4.1%	4.1%	3.9%	4.7%	4.0%	4.2%	4.1%	3.5%	4.1%	4.0%	4.3%	4.5%	4.7%	4.8%	5.6%	4.8%	1.91%	1,467
Google Maps	2.9%	1.7%	2.0%	1.8%	1.8%	1.8%	1.9%	2.2%	3.2%	4.0%	4.0%	5.0%	5.7%	5.6%	5.7%	5.8%	6.8%	6.4%	7.3%	6.6%	5.0%	4.8%	4.6%	3.4%	0.81%	1,584
Alarmclock 1	6.6%	8.2%	8.8%	10.0%	10.4%	9.7%	8.7%	6.8%	4.8%	3.3%	2.2%	1.4%	1.0%	1.0%	1.0%	1.0%	0.9%	0.6%	1.0%	1.2%	1.4%	2.0%	3.0%	5.0%	4.55%	341
Alarmclock 2	3.9%	4.3%	7.5%	9.2%	10.8%	10.7%	9.7%	9.2%	8.2%	7.7%	5.3%	3.3%	1.8%	0.5%	0.4%	0.3%	0.5%	0.6%	0.7%	0.4%	1.0%	0.6%	1.1%	2.0%	0.32%	169
Weather App	2.1%	0.9%	0.6%	0.3%	0.5%	2.0%	3.8%	2.2%	4.0%	10.1%	11.2%	9.8%	8.1%	3.2%	2.9%	5.9%	6.5%	4.3%	2.0%	4.9%	3.3%	4.0%	4.7%	2.5%	0.06%	309
Twitter	3.6%	3.3%	3.6%	4.4%	3.3%	3.0%	3.8%	4.0%	4.3%	4.3%	4.8%	4.7%	4.3%	4.4%	4.7%	4.6%	4.0%	4.1%	4.0%	4.6%	4.9%	4.8%	4.3%	4.1%	0.56%	457
Phone	2.6%	2.2%	1.9%	1.9%	1.8%	1.9%	1.9%	1.7%	2.4%	3.3%	4.1%	4.8%	5.2%	5.8%	5.7%	6.5%	6.4%	7.4%	7.6%	6.8%	5.8%	4.8%	4.1%	3.5%	1.94%	2,409
Angry Birds	5.3%	4.3%	3.2%	2.4%	1.6%	1.8%	1.5%	2.0%	1.8%	2.8%	3.5%	3.5%	4.6%	5.6%	4.9%	6.5%	4.7%	6.2%	5.6%	6.0%	5.1%	6.0%	5.4%	5.7%	0.64%	727
Kindle	9.1%	7.7%	6.9%	5.5%	4.0%	3.2%	2.9%	2.3%	1.7%	2.0%	2.3%	1.9%	2.8%	3.6%	2.6%	4.3%	2.5%	3.8%	3.1%	2.5%	4.9%	5.3%	7.3%	7.7%	0.47%	209
Calculator	2.6%	2.3%	2.0%	0.8%	0.7%	0.8%	1.1%	1.7%	2.8%	1.6%	6.6%	6.8%	7.1%	6.2%	5.6%	7.6%	7.7%	6.9%	5.8%	5.3%	6.8%	6.2%	3.3%	1.8%	0.19%	650
Calendar	5.1%	3.6%	0.7%	0.4%	0.2%	0.4%	3.8%	2.2%	3.9%	6.1%	7.6%	7.8%	6.3%	5.3%	3.5%	5.5%	5.8%	5.3%	3.8%	4.4%	5.6%	4.3%	3.1%	5.1%	0.14%	615
Camera	4.2%	4.3%	2.7%	3.5%	2.7%	2.0%	3.0%	2.2%	1.3%	2.6%	3.6%	3.2%	4.4%	5.7%	5.0%	5.1%	6.2%	6.7%	6.0%	4.7%	5.0%	5.8%	4.5%	5.7%	0.19%	781
Music	2.0%	3.6%	4.5%	5.1%	5.3%	5.4%	6.2%	6.1%	5.8%	4.5%	5.3%	3.7%	4.2%	3.8%	3.6%	3.6%	3.4%	3.6%	3.1%	4.1%	3.6%	3.8%	2.9%	2.8%	0.41%	483

Figure 10. Application usage time throughout the day. Within each row (i.e., for each app) low usage is indicated by white, increasing through yellow and reaching a peak at red. Percentages indicate the usage time of each app and are normalized within each row.

The current design of the *AppSensor* is not capable of tracking multitasking. For instance, if a user is listening to music – with the player running in the background from the operating system's perspective – and is browsing the Internet at the same time, the *AppSensor* will return the browser as the open application. Similarly, on the Android platform we have the problem that applications' widgets are part of the home screen application. Therefore we cannot measure the widget-based usage of apps. However, most widgets are simply entry points into apps.

While we have no detailed information on the participants due to the domain of the underlying platform *appazaar* – i.e. supporting people to find new apps – we may assume that some of our users are early adopters with a high affinity to apps. Thus, our participants in general may have a slightly higher affinity toward app usage than the general population.

Like every sensor, the *AppSensor* is not error-free. For instance, it might return values that do not relate to the user's current activity. A user might leave and put away the device with an app still running. The uncertainty of the reasoned context will increase with the time that the user has not used her device. However, most devices go into standby after some time of non-usage, as long as the user does not intentionally use an app that prevents standby. Moreover, app usage that occurs when standby mode is purposefully disabled can also be valued as valid usage.

Furthermore, the *AppSensor* cannot be used to reason on a user's context when no application is used at all, i.e. in device standby and turned off. Secondly, the sensor is obviously only available during active usage of the device. Otherwise it can only be deduced that the user is currently not using his device. Thirdly, the *AppSensor's* accuracy also depends on its sample rate. This impacts the quality of the measured data. The sample rate needs to be chosen depending on how often a user is switching between different applications. If the swapping frequency is higher than the sample rate, the accuracy will decrease. However, at a high frequency the system load might increase and impact power consumption. We believe our sample rate is correctly positioned given these constraints, as we conducted an informal pre-study on how fast one can start a new app.

Of course, our findings cannot be transferred to general usage of the underlying services. For instance, it might be the case that people use Facebook during the day on their stationary PC or laptop, and use their mobile device when they are lying in bed in the evening.

Whether or not an *AppSensor* is widely deployable within a system strongly depends on the underlying operating system and the policies of the device's vendor. The *AppSensor* used in this paper was implemented on the Android platform because Android provides the required openness. The sensor itself needed to be implemented as background service, which is not possible on every device. For these and other reasons an *AppSensor* is not possible on Apple's iPhone, or at least cannot be deployed in the wild.

CONCLUSION

In this paper, we conceptualized and studied the *AppSensor*: a framework for analyzing smart phone application usage. For the first time – to the best of our knowledge – the method of deployment-based research by means of app store deployments was combined with fine-grained data collection on mobile application usage. In contrast to physical sensors (e.g. GPS for locations), we defined a virtual sensor for measuring the usage of mobile applications. This public deployment of *AppSensor* provided us with the data of more than 4,100 users over a period longer than four months. In short, this paper included the following contributions (amongst others):

- a descriptive analysis of our sample data showing that (among other findings) mobile device users spend almost an hour a day using apps but spend less than 72 seconds with an app at a time (on average), and that average usage time differs extensively between app categories,
- a context-related analysis that led to the following conclusions (among other findings): (1) mobile phones are still used mostly for communication (text and voice); (2) some apps have somewhat intense spikes in relative usage (e.g. music and social apps), whereas others are more broadly employed throughout the day; (3) when people actively use their devices they spend less time with each app; (4) short sessions with only one app are much more frequent than longer sessions with two or more apps, and the first

app within a session is very likely to be an app for communication; (5) when people are traveling they are more likely to use multimedia apps and they are surprisingly less likely to use travel apps,

- the conceptual design of our research method, namely the *AppSensor* as a virtual sensor for measuring mobile app usage.

We believe that the MobileHCI community should be aware of this data set. Therefore it is our plan to make the whole data set available to the community[7], allowing others to draw their own conclusions and perform their own analysis that may go beyond what we have found in the data. To our knowledge this is the first attempt to analyze application usage at this scale and we believe that our work provides data to verify and deepen findings of the sort that Demieux and Losguin [8] and Verkasalo [18] have presented in their previous but smaller studies.

For future work, we will use the findings of this paper to further inform the design of the *appazaar* recommender system. The chain of previously used apps will provide much information about users' tasks and intentions. Developing models that are able to predict the next-to-be-used app will dramatically increase the usefulness of an app recommender system. We are also working to better understand our location-based results with more detailed spatial analysis.

[7]Access to the anonymized data set is available upon request.

REFERENCES

1. Adomavicius, G., and Tuzhilin, A. Context-Aware recommender systems. In *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Eds. Springer US, Boston, MA, 2011, ch. 7, 217–253.
2. AppsFire.com. Infographic: iOS Apps vs. Web Apps. `http://blog.appsfire.com/infographic-ios-apps-vs-web-apps`, accessed on Feb. 15, 2010.
3. Barkhuus, L., and Polichar, V. Empowerment through seamfulness: smart phones in everyday life. *Personal and Ubiquitous Computing* (Dec. 2010), 1–11.
4. Böhmer, M., Bauer, G., and Krüger, A. Exploring the Design Space of Recommender Systems that Suggest Mobile Apps. In *Proc. of Workshop CARS '10* (2010).
5. Böhmer, M., Prinz, M., and Bauer, G. Contextualizing Mobile Applications for Context-aware Recommendation. In *Adj. Proc. of Pervasive '10* (2010).
6. Church, K., and Smyth, B. Understanding mobile information needs. In *Proceedings of the 10th international conference on Human computer interaction with mobile devices and services*, MobileHCI '08, ACM (New York, NY, USA, 2008), 493–494.
7. Cui, Y., and Roto, V. How people use the web on mobile devices. In *Proceeding of the 17th international conference on World Wide Web*, WWW '08, ACM (New York, NY, USA, 2008), 905–914.
8. Demumieux, R., and Losquin, P. Gather customer's real usage on mobile phones. In *Proceedings of the 7th international conference on Human computer interaction with mobile devices and services*, MobileHCI '05, ACM (New York, NY, USA, 2005), 267–270.
9. Dey, A. K. Understanding and using context. *Personal and Ubiquitous Computing 5*, 1 (Feb. 2001), 4–7–7.
10. Froehlich, J., Chen, M. Y., Consolvo, S., Harrison, B., and Landay, J. A. Myexperience: a system for in situ tracing and capturing of user feedback on mobile phones. In *Proceedings of the 5th international conference on Mobile systems, applications and services*, MobiSys '07, ACM (New York, NY, USA, 2007), 57–70.
11. Girardello, A., and Michahelles, F. Appaware: which mobile applications are hot? In *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*, MobileHCI '10, ACM (New York, NY, USA, 2010), 431–434.
12. Henze, N., Poppinga, B., and Boll, S. Experiments in the wild: public evaluation of off-screen visualizations in the android market. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*, NordiCHI '10, ACM (New York, NY, USA, 2010), 675–678.
13. Herlocker, J. L., Konstan, J. A., and Riedl, J. Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*, CSCW '00, ACM (New York, NY, USA, 2000), 241–250.
14. Kaasinen, E. User needs for location-aware mobile services. *Personal and Ubiquitous Computing 7*, 1 (May 2003), 70–79.
15. Kray, C., and Rohs, M. Swiss Army Knife meets Camera Phone: Tool Selection and Interaction using Visual Markers. In *Proc. of Joint Workshops MIRW '07 and MGuides '07* (2007).
16. McMillan, D., Morrison, A., Brown, O., Hall, M., and Chalmers, M. Further into the wild: Running worldwide trials of mobile systems. In *Pervasive Computing*, P. Floréen, A. Krüger, and M. Spasojevic, Eds., vol. 6030 of *Lecture Notes in Computer Science*. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2010, ch. 13, 210–227–227.
17. Satyanarayanan, M. Swiss army knife or wallet? *IEEE Pervasive Computing 4*, 2 (Apr. 2005), 2–3.
18. Verkasalo, H. Contextual patterns in mobile service usage. *Personal and Ubiquitous Computing 13*, 5 (2009).