嗨Wency，这篇研究论文的重点是使用移动应用程序时的中断。关于检测中断和测量中断的方法，还有一个有趣的部分，您可能会发现它很有用。

注释＃5和＃6似乎是摘要的有用描述。
＃20，＃22和＃28至＃30注释似乎与您的兴趣非常接近。
注释＃26之后的信息为从中断中恢复提供了有趣的建议。

标题：返回应用程序：移动应用中断的成本
文件：Back_to_the_app_annotated.pdf
资源: https://dl.acm.org/citation.cfm?id=2371617

注解
这些项目符号上的数字对应于添加到文档中的数字。

1. 在有意或无意的情况下，智能手机用户与应用程序交互时可能会被打断
2. 移动应用程序中断可以分为两种情况：（1）在应用程序之间进行有意的来回切换；（2）传入电话引起的意外中断
3. 中断可能会带来大量开销（可能会使任务完成延迟最多4倍）
4. 智能手机的作用已从纯粹的通讯设备变为多功能工具集。
5. 到目前为止，移动应用程序支持的功能数量正在稳步增长。同时，由于智能手机的功能有限以及制造商施加的安全限制，当前用于智能手机的操作系统缺少各个应用程序之间相互作用的可能性。因此，用户不必使用单独的任务专用设备，而必须在应用程序之间进行切换。从根本上讲，当发生意外事件（例如电话）时，应用程序将中断。
6. 从交互的角度来看，移动化的成本很高。尽管任务中断是智能手机中固有的问题，但对于应用程序的细粒度使用这种现象知之甚少。但是，仍然存在一个悬而未决的问题，这种中断多久出现一次，它们对用户的性能有何影响，尤其是它们对任务完成时间造成了多少成本。
7. 移动应用程序中断可以分为两种情况：（1）在应用程序之间进行有意的来回切换；（2）传入电话引起的意外中断
8. 在本研究中，任务完成是使用应用程序所花费的时间
9. 对移动设备的持续关注分散了，主要是由于环境干扰，并分解为较短的时间跨度。
10. 方法。中断检测。如图1所示，当前台活动从应用程序x更改为y（x = y）然后返回到x时，将考虑中断。
11. 内部中断是由用户引起的。外部中断是由需要立即处理的传入应用程序引起的，例如电话
12. 有关如何测量用户干扰的信息
13. 处理日志后的数据集摘要
14. 每个用户（前2行）和每个应用的平均值。

15. 与用户通过电话打扰（外部打扰）相比，内部打扰更为频繁并且涉及更多的应用，如表3的前两行所示。这些差异在统计上是显着的。
16. 总体而言，外部中断（通过电话）完成的时间比内部中断要多
17. 与内部模式相比，电话呼叫在中断的应用程序上产生了显着更高的开销时间
18. 这些结果表明，人们在使用应用程序时通常会更加投入。返回中断的任务时，外部中断更具破坏性。
19. 在应用程序级别的移动中断很少发生：在日常通话中，大约每天使用率的3%，在应用程序之间来回切换时占8%。但是，这样做的话，恢复成本可能会非常高。
20. 中断可能是设计智能手机交互的有用方法。同样，了解中断的持续时间和由此产生的开销可以帮助改善旨在或多或少地支持多任务的应用程序的设计。
21. 对于从桌面中断中恢复，有两个建议：提醒用户未完成的任务，并帮助他们有效地调用任务上下文。
22. 在移动域中，关键因素是减少应用程序中断带来的开销时间。
23. 当任务中断发生时，用户可以准备离开当前任务。例如，对于来电，呼叫者通常在线路上等待几秒钟。
24. 这样，用户将能够保存精神状态，并在被中断之前牢记最近中断的应用程序。
25. 我们相信，将此通知逐渐覆盖到当前使用的应用程序上还将为用户提供对他最近的动作进行潜意识快照的可能性。
26. 当用户恢复先前中断的应用程序时，她必须重新分配认知资源，如果从一开始对资源的需求很高，则这将变得越来越困难
27. 任务和应用程序之间的直接映射很难传达，因为任务可能涉及单个应用程序（例如，阅读文档）或一系列应用程序（例如，为开会做准备，用户需要检查日历，咨询网页并做笔记）。
28. 应用程序中断很少在智能手机上发生，但一旦中断，对用户而言确实会造成巨大的损失。这给移动设计师和智能手机供应商带来了许多新挑战。
29. 应用切换行为的发生频率不像预期的那样高。尽管智能手机允许更改交互的焦点，但用户不愿意这样做。原因之一可能是，在移动应用程序之间进行切换之后，尚无任何机制或合适的交互技术来支持重新获得上下文。
30. 与应用程序切换相比，电话中断会给被中断的应用程序增加相当大的开销。这是预料之中的，因为来电可能随时发生。但是，令人惊讶的是注意到中断应用程序的成本：高达四倍。
31. 相关研究论文

Back to the App: The Costs of Mobile Application Interruptions

Luis A. Leiva[1,*] **Matthias Böhmer**[2] **Sven Gehring**[2] **Antonio Krüger**[2]
[1]ITI/DSIC, Universitat Politècnica de València (Spain) [2]DFKI GmbH, Saarbrücken (Germany)
llt@acm.org, {matthias.boehmer, sven.gehring, krueger}@dfki.de

ABSTRACT
Smartphone users might be interrupted while interacting with
1 an application, either by intended or unintended circumstances. In this paper, we report on a large-scale observational study that investigated mobile application interruptions in two scenarios: (1) intended back and forth switching be-
2 tween applications and (2) unintended interruptions caused by incoming phone calls. Our findings reveal that these interruptions rarely happen (at most 10% of the daily application usage), but when they do, they may introduce a significant
3 overhead (can delay completion of a task by up to 4 times). We conclude with a discussion of the results, their limitations, and a series of implications for the design of mobile phones.

Author Keywords
Interruptions; Application Switching; Task Interleaving; Task Deferral; Resumption Lags; Large-scale Study

ACM Classification Keywords
H.5.m. Information Interfaces and Presentation (e.g. HCI)

INTRODUCTION
Nowadays, smartphones have emerged as multipurpose devices. Besides calls and text messages, smartphones offer the possibility of installing multiple applications for a variety of purposes, e.g., gaming, browsing, listening to music, editing pictures, etc. Hence, the role of smartphones has
4 changed from pure communication appliances to multifunctional toolsets.

So far, the amount of functionalities that are supported by mobile applications is steadily increasing. At the same time, due to the limited capabilities of smartphones and the security restrictions imposed by the manufacturers, current operating
5 systems for smartphones lack possibilities for interplay between individual apps. As a result, instead of using a separate task-dedicated appliance, users have to switch between applications; and more elementally, applications are interrupted when unintended events like phone calls occur.

From the perspective of interaction, being mobile is cognitively costly [5]. Although task interruption is an inherent problem in smartphones, little is known about the phenomenon at the fine-grained level of application usage. 6
Yet it remains an open question how often such interruptions appear, what impact they have on the user's performance and, especially, what costs they impose on task completion times.

Researchers have investigated task interruption in the desktop domain (see next section). In this paper, we extend findings of previous in-lab studies to the domain of smartphones. We conducted a large scale observational study "in the wild" that investigated mobile application interruptions in two scenarios: (1) intended back and forth switching between applica- 7
tions, and (2) unintended interruptions caused by incoming phone calls. Concretely, we looked into the cost of these interruptions on task completion time. Therefore, in the context of this work, and acknowledging its limitations, we consider task completion as the time spent using an application. 8

We present a framework that can easily assist researchers to detect application interruptions on mobile phones at a large scale. Our findings show that interruptions caused either by incoming phone calls or by application switching rarely happen on smartphones but, as soon as they do, they might cause a significant impact on the interrupted activity. We conclude with a discussion on preventive and curative strategies, and their implications for the design of mobile phones.

RELATED WORK
Task interruptions have been extensively studied on stationary PCs (see, e.g., [1, 3, 4, 7, 8]), but little has been reported for mobile users and their unique set of difficulties. It is clear that mobility imposes cognitive restrictions and continuous interruptions on application usage; however, to the best of our knowledge, there is no previous research that focuses *exclusively* on the application level. Moreover, other studies in a similar vein [5, 6] have been performed in carefully controlled settings. Our main aim is to investigate the costs of mobile application interruptions "in the wild", i.e., in a natural, general environment *and* at scale.

Oulasvirta et al. [6] carried out a study (28 subjects) on mobile Web search tasks while moving, and observed that continuous attention to the mobile device is fragmented, mostly
due to environmental distraction, and broke down into short 9
time spans. Karlson et al. [5] focused on the tasks as a whole, including switching to a PC, if necessary, to complete them. They carried out a 2-week diary study mostly focused on email management (24 subjects), characterizing how problematic interruptions are to mobile users and identifying primary sources of frustration. In addition, Böhmer et al. [2]

---
[*]Work done while visiting the DFKI institute.

MobileHCI'12, September 21–24, 2012, San Francisco, CA, USA.

looked into unconstrained mobile application usage, reporting descriptive statistics on what is probably one of the largest datasets in mobile HCI today. However, they did not study how often interruptions happen and what their real impact is on the user's productivity.

STUDY

We analyzed the *AppSensor* dataset provided by Böhmer et al. [2]. *AppSensor* is an Android background service that indicates the currently used (foreground) application at a sampling rate of 2 Hz. It also provides data related to phone usage such as runtimes or timestamps of screen on/standby. *AppSensor* collapses measured values into single data samples, with one sample denoting which application was launched at what point in time and for how long it was used; e.g., the phone application was launched at 6:30 PM and then was used for 32 minutes. A detailed description of this corpus as well as the *AppSensor* tool can be found in [2].

The dataset contained around 5.5M data samples, corresponding to 15.7K different applications used for one year and a half (from August 2010 to January 2012, Table 1) by 3.6K unique users who were geographically distributed worldwide.

Data samples	Days of study	Applications	Users
5,495,815	532	15,756	3,611

Table 1: Features of the analyzed dataset.

Method

Detecting Interruptions

10 As observed in Figure 1, an interruption is considered when the foreground activity changes from application x to y ($x \neq y$) and then returns to x. Additionally, as indicated in the figure, we impose the following restrictions: *1)* Launcher/dialer applications (L) and calls (C) are not considered to be interrupted. *2)* An application cannot be interrupted by L. More formally, $L \neq x \neq C$ and $y \neq L$. This way, applications can be deferred (i.e., *interrupted on purpose*) or interrupted by an
11 *incoming* phone call. For the sake of brevity, we will refer to the former case as **internal** interruptions, and will use **external** for the latter.

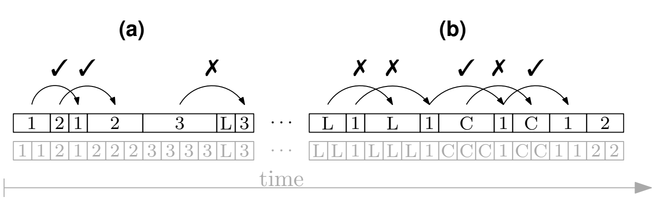

Figure 1: Light-gray boxes illustrate app identifiers from the data samples, which were used to recompose the sequences of each user's activity (black boxes) and detect **internal** [1a] and **external** [1b] interruption patterns.

It is worth noting that applications can be deferred for a long time because of environmental distractions not related to mobile phone usage (e.g., being prompted by a friend to chat for a while), in which case the device switches to stand-by mode. Because of this, to avoid misleading results, we use a time window of 1 minute; so that if a series of consecutive data samples were found for the same user + day + app, such application was considered as a different activity if the time between data samples exceeded 1 min.[1] We chose this specific duration for the time window based on the average application usage length according to the literature [2, 10].

Measuring Interruptions

Let T_n be the runtime of an application in *normal* operating 12
condition (i.e., without interruptions) and let $T_r = T_b + T_a$ be the runtime of the same application when it is interrupted, which is decomposed into the runtime *before* the interruption T_b and the runtime *after* the interruption T_a (until the application is closed, or another interruption is detected). As shown in Figure 2, $T_o = T_r - T_n$ is the overhead imposed on the application as a consequence of the interruption T_i. Overheads are also cited as "resumption lags" in the literature [3, 8], and usually lead to a decrease in primary-task performance. However, notice that, while $\{T_a, T_b, T_i, T_n\} \in \mathbb{R}_0^+$, $T_o \in \mathbb{R}$, i.e., overheads can be either positive or negative, since an application that is used in normal conditions for a very long time might not be resumed when the user recovers from an interruption, yielding $T_r < T_n$ and hence $T_o < 0$.

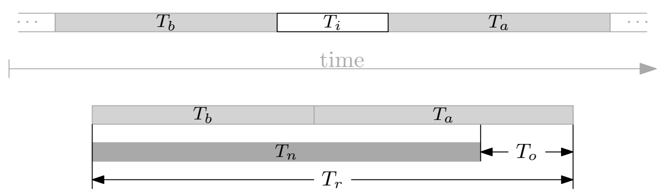

Figure 2: Computing an application overhead when interrupted.

Notice also that overheads can only be fairly computed for paired cases, i.e., one needs to compare the time an application is interrupted against its normal usage time for a given user. Nonetheless, while mining the dataset we computed all possible cases (including unpaired conditions), in order to quantify precisely how often interruptions did happen (Table 2). Then, unpaired cases were filtered and dropped from the subsequent analysis (Table 3 and Results section).

Procedure

Logs were sequentially grouped per day per user. Then we processed all interaction sessions, which are summarized in Table 2, and computed a series of descriptive statistics, depicted in Table 3. These values were macro-averaged, to give equal weight to each user and their applications. Outliers were considered when the mean exceeded 1.5 the interquantile range. To highlight the differences between both types of interruptions, we carried out different hypothesis tests. Interaction effects were considered at the $p < .05$ level.

RESULTS

As previously mentioned, we report here the results of our study for **external** vs. **internal** interruptions after removing

[1]Concretely when $t_j - (t_{j-1} + r_{j-1}) > 1\text{min}$, where t_j is the jth log timestamp and r_{j-1} is the application runtime.

	external	internal
Interruption data samples	776,922	970,543
Interrupted users	1,929 [1,676]	2,926 [2,609]
Interrupted applications	1,373 [487]	4,626 [1,043]

13 **Table 2:** Dataset summary after processing the logs, showing in brackets the number of balanced (paired) cases.

outliers and unpaired cases (final sample sizes are denoted in brackets in Table 2).

	external	internal
Daily interruptions (% usage)	3.2 (2.2)	8.3 (5.3)
Interrupted applications	3.3 (2.6)	8.7 (7.2)
Regular app. runtime (s)	24.8 (31.8)	18.9 (24.4)
Runtime when interrupted (s)	107.1 (121.1)	87.9 (75.5)
Interruption duration (s)	12.5 (8.1)	23.7 (19.3)
Overhead duration (s)	43.2 (65.9)	34.4 (40.7)

14 **Table 3:** Mean (and SD) values per user (first 2-rows) and per app.

Since data could not be considered as normally distributed[2], we used the Kolmogorov-Smirnov test, which is non-parametric[3] and hence does not make assumptions about data distribution.

15 Unsurprisingly, **internal** interruptions are more frequent ($D^+ = 0.50, p < .0001$, Cohen's $d = 1.24$) and involve more applications ($D^+ = 0.36, p < .0001, d = 0.98$) in comparison to when users are interrupted by phone calls, as shown in the first 2 rows of Table 3. These differences were found to be statistically significant.

16 Overall, applications interrupted by phone calls take more time to complete than switched applications ($D^+ = 0.06, p = .04, d = 0.18$). Notice the differences on application runtime when compared to its normal usage ($D = 0.09, p = .003, d = 0.20$). The duration of phone calls was found to be significantly smaller than the duration of the switched applications ($D^- = 0.31, p < .0001, d = 0.75$), however phone calls produce a significantly higher overhead
17 time on the interrupted application compared to the **internal** pattern ($D^- = 0.08, p = .006, d = 0.17$). These results
18 suggest that people usually engage more when using applications, and that **external** interruptions are more disruptive regarding task recall.

We computed the correlations between these measurements, and they were found to be mostly weak/moderate (Figure 3). The most consistent correlations found both in the **external** and **internal** groups were, as expected, those of *1)* the runtime of an interrupted application vs. its overhead; and *2)* number of daily interruptions vs. number of interrupted applications.

19 In sum, it was interesting to observe that, overall, mobile interruptions at the application level do not happen often: around 3% of daily usage in case of phone calls, 8% when

[2]Verified by previous Shapiro-Wilk tests ($p < .0001$ in all cases).

[3]Kolmogorov's D statistic refers to two-tailed comparisons, while D^+ and D^- refer to one-tailed comparisons.

	n_i	n_a
n_i	–	0.26***
n_a	0.27***	–

(a) Per user.

	T_r	T_i	T_o
T_r	–	0.12**	0.68***
T_i	0.09*	–	0.13**
T_o	0.51***	0.22***	–

(b) Per user applications.

Figure 3: Correlation study. Above main diagonal: ρ for **external** interruptions. Below main diagonal: ρ for **internal** interruptions. [3a] n_i: number of daily interruptions, n_a: number of interrupted applications. [3b] T_r: interrupted application runtime, T_i: interruption time, T_o: overhead time. Statistical significance: $^*p < .05$, $^{**}p < .01$, $^{***}p < .001$.

switching back and forth between applications; but, when they do, the resumption cost may be exceedingly high.

DISCUSSION

The extent to which an interruption occurs can be a helpful means for designing smartphone interaction. Also, knowing 20 both the duration of the interruption and the resulting overhead can help to improve the design of applications that are aimed to support multitasking to a greater or a lesser degree.

An obvious approach to reach these goals is helping the user regain the context of the deferred application when it is resumed. On the desktop arena, when this happens, pertinent visual cues are given as a help for easing the recovery from the interruption; however, this is usually not applied to smartphones [5]. Iqbal and Horvitz [3] offered two directions for recovering from interruptions on the desktop: reminding users of unfinished tasks and assisting them in efficiently re- 21 calling task context. We believe that, in the mobile domain, the key factor is reducing the overhead time that is introduced by an application interruption. To do so, we suggest either helping the user to maintain the context while switching to 22 another application, or to support regaining context when returning to the interrupted application. Next, we describe design considerations that can support these recommendations.

Design Implications

Our results entail a series of interventions for designing mobile interaction to reduce the overhead that is introduced by application interruptions. In general, inspired by previous approaches in the field of desktop interruptions, we distinguish between *preventive* (preparing the user for being interrupted, cf. [9]) and *curative* (supporting the user after being interrupted, cf. [3]) strategies.

Preventive: Preparation for Being Interrupted

When a task interruption occurs, the user could be prepared to leave the current task. For instance, for incoming phone 23 calls the caller usually waits on the line for some seconds. Postponing the call a bit more (say, 500 ms) might provide time to give the user an auditory/visual/haptic signal that soon the phone application will pop-up. This way, the user would be able to save a mental state and keep in mind the recently 24 interrupted application before he is interrupted. Further, currently users receive a full-screen visual notification of incoming calls on most smartphones. We believe that gradually overlaying this notification onto the currently used applica- 25 tion would also provide the user with the possibility to take a subconscious snapshot of his most recent action. This particular approach may apply equally to the **internal** interruptions.

Curative: Guidance for Going Back into Tasks

26 When the user resumes a previously interrupted application, she has to reallocate cognitive resources, which becomes increasingly difficult if the resource demands were high to begin with [3]. Therefore, she should be given some help to be able to immediately (and easily) continue with it. For example, this could be achieved by automatically replaying the last N milliseconds of UI interactions, to give a hint of what she was doing before the interruption. The system could also leave a visual on-screen cue such that the user could remember at any time to which task she is switching back. Alternatively, when returning to the interrupted application, the canvas of the foreground application could vanish into the direction of the last focus of interaction, in order to guide the user to the screen position before the interruption took place.

Limitations of the Study

First, our results are dependent on the quality of the dataset. As previously pointed out, there exist many disruption contenders apart from mobile application usage itself that lead people to change tasks and applications [1]. In addition, while the overhead measure has been used in the literature for routine tasks, it is ideally suited to resumption where there is little re-encoding of the primary task required (see [7]). These facts and other environmental factors are an important source of indeterminism that we were unfortunately not able to measure within the data. Although we can report that interruptions *do* happen at the application level and that their cost is significant, our observational study was non-controlled, based on pure data mining. Hence, a line of future work will be enriching the data collection process with additional experience sampling approaches. This would provide us with deeper insights about tasks being interrupted.

27 Furthermore, we acknowledge that a direct mapping between tasks and applications is hard to convey, as a task can involve a single application (e.g., reading a document) or many (e.g., to prepare a meeting one can usually check the calendar, consult a web page, and take some notes). As such, it is not clear to what extent the ecological impact of mobile application interruptions would align with user's cognitive load and higher-level goals; e.g., one cannot guarantee if a 2-minute phone call would really significantly interrupt a user preparing a meeting.

Finally, another observation associated with this analysis is that, besides having used a large population with different backgrounds in a realistic context, the analyzed user sample comes from one of the mainstream mobile ecosystems. As such, further research would be needed to understand application interruptions in other platforms and verify whether our findings still hold.

CONCLUSION

In this paper, we replicated findings of previous lab studies on desktop interruptions and extended them to the mobile domain, by exploiting a large-scale dataset of mobile application usage. We have observed that app-switching behavior as well as incoming phone calls are a non-negligible source of disruption and therefore should be mitigated. Our study reveals three general findings:

1. Application interruptions rarely happen on smartphones, but when they do, they can be really costly for the user. 28
   This poses a wealth of new challenges for mobile designers and smartphone vendors.
2. App-switching behavior does not happen as often as it is presumed. While smartphones allow changing the focus 29
   of interaction, users are reluctant to do so. One reason might be that there are no mechanisms or suitable interaction techniques (yet) to support regaining context after switching between mobile applications.
3. Phone call interruptions add a significantly high overhead on the interrupted application in comparison to those of 30
   app-switching. This was expected, as incoming calls potentially can happen anytime. However, it was surprising to note the cost on the interrupted application: the runtime could be increased by up to four times.

Furthermore, we have discussed possible approaches to reduce the overhead caused by application interruptions and to help users resume task flow. Aside from the limitations of the study, it is our belief that our observations can lead to helpful guidelines for mobile interaction design.

ACKNOWLEDGMENTS

We thank Duncan Brumby and our reviewers for their valuable suggestions to strengthen this paper. We also thank Antti Oulasvirta for providing initial feedback. This work is partially supported by the MIPRCV project (CSD2007-00018).

REFERENCES

1. Benbunan-Fich, R., Adler, R. F., and Mavlanova, T. Measuring multitasking behavior with activity-based metrics. *ACM TOCHI 18*, 2 (2011), 7:1–7:22.
2. Böhmer, M., Hecht, B., Schöning, J., Krüger, A., and Bauer, G. Falling asleep with Angry Birds, Facebook and Kindle – A large scale study on mobile application usage. In *Proc. MobileHCI* (2011), 47–56.
3. Iqbal, S. T., and Horvitz, E. Disruption and recovery of computing tasks: field study, analysis, and directions. In *Proc. CHI* (2007), 677–686. 31
4. Jin, J., and Dabbish, L. A. Self-interruption on the computer: a typology of discretionary task interleaving. In *Proc. CHI* (2009), 1799–1808.
5. Karlson, A. K., Iqbal, S. T., Meyers, B., Ramos, G., Lee, K., and Tang, J. C. Mobile taskflow in context: A screenshot study of smartphone usage. In *Proc. CHI* (2010), 2009–2018.
6. Oulasvirta, A., Tamminen, S., Roto, V., and Kuorelahti, J. Interaction in 4-second bursts: the fragmented nature of attentional resources in mobile HCI. In *Proc. CHI* (2005).
7. Salvucci, D. D. On reconstruction of task context after interruption. In *Proc. CHI* (2010), 89–92. 31
8. Salvucci, D. D., and Bogunovich, P. Multitasking and monotasking: The effects of mental workload on deferred task interruptions. In *Proc. CHI* (2010), 85–88.
9. Trafton, J. G., Altmann, E. M., Brock, D. P., and Mintz, F. E. Preparing to resume an interrupted task: effects of prospective goal encoding and retrospective rehearsal. *Int. J. Hum.-Comput. Stud. 58*, 5 (2003), 583–603. 31
10. Vaz De Melo, P. O. S., Akoglu, L., Faloutsos, C., and Loureiro, A. A. F. Surprising patterns for the call duration distribution of mobile phone users. In *Proc. ECML PKDD* (2010), 354–369.