Research in the App Store Era: Experiences from the CenceMe App Deployment on the iPhone

Emiliano Miluzzo, Nicholas D. Lane, Hong Lu, Andrew T. Campbell
CS Dartmouth College
Hanover, NH, USA

ABSTRACT
Smartphones and "app stores" are enabling the distribution of a wide variety of third party applications to very large numbers of people around the globe in an instance with the potential to collect rich, large-scale data sets. This new era represents a game changer for our research community - one which we are still trying to best understand and exploit. We discuss our experiences in developing, distributing and supporting CenceMe, a personal sensing application for mobile social networks, developed for the Apple iPhone and first released when the Apple App Store opened in 2008. We had to come to terms with supporting a fairly complex real-time sensing application outside the normal controlled laboratory setting. Instead of deploying the CenceMe application to a small set of local users (e.g., 30+ users when we first deployed CenceMe on Nokia N95s in 2007) we had to deal with thousands of users distributed around the world. This new era of app development and distribution at scale is an exciting one for researchers - one that will accelerate the deployment of new ideas and likely lead to new advances and breakthroughs not well understood today.

Author Keywords: Mobile Phone Sensing, Mobile Sensing Systems, Large-Scale Sensing Deployments.

ACM Classification Keywords: H.0 Information Systems (General): Miscellaneous. **General Terms:** Experimentation, Human Factors, Measurement, Performance.

INTRODUCTION
The smartphone's programmability, pervasiveness and growing computational capability, along with the application distribution systems support (i.e., vendor specific app stores) are contributing to an explosion of smartphone-centered research across many area of interest to the UbiComp community: from gaming to social networking, green applications, and mobile sensing. The application distribution system support in particular (e.g., Apple App Store, Android Market, and Nokia Ovi) is a game changer for the research community because it enables the deployment of applications to millions of smartphones in the blink of an eye and gives the opportunity to collect very large data sets from the wild as never possible before. By mining rich, large-scale data sets researchers will be able to answer novel and exciting research questions discovering, for example, the way people use and interact with their mobile phones [5, 6], with location-based social networks [8] and green apps [4].

In this paper, we report on our experience from releasing CenceMe [10] to the public through the Apple App Store. CenceMe is a social sensing application for smartphone that was first implemented on the Nokia N95 devices in 2007 and evaluated as part of a small scale study [10]. Following this, CenceMe was ported to the iPhone and released publicly in July 2008 when the App Store was first launched. Since its release on the App Store CenceMe has been used by over 9100 active users in over 95 countries. The goals of the iPhone CenceMe release are the following: i) scale our project outside the confined settings of a research lab and give the application a "global" dimension; ii) understand how a mobile sensing application works in the wild without the direct control of the researchers; iii) assess the interest of people toward a mobile sensing application, which, by exploiting the phone's sensors to transparently infer human activity and the surroundings of the user, opens up new dimensions, in terms of content and privacy.

We describe the design of the iPhone CenceMe client and backend and the challenges encountered in building a mobile sensing application to be publicly used. We discuss the "*deploy-use-refine*" approach we adopt in order to evolve the system design as user feedback is collected. We describe the software challenges and limitations that could impact the capabilities of a global mobile sensing application. When a mobile sensing application runs unmanned in the wild there is no guarantee that the event being inferred (having sensed it using the phone onboard sensors) is correct, since there is no notion of ground truth. We propose a mechanism that boosts the fidelity of mobile sensing applications by relying on a multi-sensing modality approach to mitigate the effect of lack of ground truth data.

It is important to point out that releasing an application to the public for research purposes requires to be compliant with the ethic code, privacy, and security constraints that protect the users of the application. It is also necessary to follow the research institution or university ethics recommendations from the IRB (Institutional Review Board) regulating user data collection and treatment.

IPHONE CENCEME
CenceMe is a social sensing application which integrates with popular social networking platforms such as Facebook,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
UbiComp '10, Sep 26-Sep 29, 2010, Copenhagen, Denmark.
Copyright 2010 ACM 978-1-60558-843-8/10/09...$10.00.

MySpace, and Twitter to augment a person's context status by using the phone's onboard sensors [10]. By running machine learning algorithms on the mobile phone itself over the sensor data collected by the phone's accelerometer, microphone, bluetooth/wifi/GPS, and camera and fusion techniques on backend machines, CenceMe infers a person's activity (e.g., sitting, walking, running) and social context (e.g., dancing, with friends, at a restaurant, listening to music) in an automated way. This information is shared within the person's social circle automatically giving the user the ability to customize their privacy settings to regulate what and where to publish the sensing-based presence. In what follows, we describe the architecture of the iPhone CenceMe client (in order to meet usability, classifiers resiliency, and preserve the phone user experience in terms of battery life for example) and backend (designed to be robust against failures, bursty user access, etc).

Client

The iPhone CenceMe client is implemented using a combination of Objective-C and legacy ANSI C code. Objective-C is mainly used for the user interface implementation, to access the low level sensors, the internal sqlite database, and to respect the model-view-controller principle of iPhone OS. C is adopted to implement the activity recognition classifier (which relies on a decision tree algorithm), and for the audio classifier (that determines the surrounding audio level - noise, quiet - or if a person is in a conversation). The audio classifier is a support vector machine (SVM) technique using the LIBSVM C library [3].

The client is responsible for: i) operating the person's presence inference over the sensor data by locally running the inference algorithms; ii) communicating the inference labels to the backend; iii) displaying the user's and their buddies sensing presence (activity, social context, location), the privacy configurations, and various other interface views that allow, for example, a user to post short messages on their preferred social network account. The classifiers are trained offline in a supervised manner, i.e., taking large collections of labeled data for both the audio and accelerometer modalities and using that data to train the classification models which are later deployed on the phone. Although earlier versions of the iPhone OS did not support multitasking (i.e., the capability to run applications in the background) the CenceMe client is designed to properly duty-cycle the sensing, inference routines, and communication rendezvous with the backend to limit the battery drain of the phone when the application is active. Reducing the battery drain is critical to avoid rapid battery discharges when the application is used, a condition that would negatively impact the phone user experience.

Cloud

The iPhone CenceMe backend, which is implemented on the Amazon Web Service cloud infrastructure [1], is comprised by a series of different virtual machine images. Each machine is an Amazon EC2 virtual machine instance running Linux which provides a series of PHP and Python based REST web service allowing multiple machines to be composed together. Each image performs a different role in the backend infrastructure and has been designed to be initialized and composed together to offer different operating points of cost and performance. This allows us to temporarily initialize different numbers of machines of different types depending on the existing or expected user workload. It also allows us to manage the cost of running the CenceMe service so that we can provision additional machines (which incur additional costs) only when user demand requires it (for example, when a new model of the Apple iPhone is released and temporarily many users try out our application, which causes us to reconfigure our system).

The server side system is responsible for: i) storing user sensor presence information and allowing other CenceMe clients restricted access to this data; ii) publishing this sensor presence information to third party social network such as Twitter and Facebook; iii) maintaining the CenceMe social network friend link structure; iv) performing routine user registration and account maintenance tasks; and v) the collection of statistics about user behavior both on the client and backend side of the system.

Figure 1. The "*deploy-use-refine*" model adopted in CenceMe.

LESSONS LEARNT

In this section, we discuss the experience we gained by deploying CenceMe on the App Store and having it used by thousand of users worldwide. Throughout the development and the evolution stages of iPhone CenceMe we applied a "*deploy-use-refine*" model (see Figure 1). According to this strategy, after initial tests in the lab, the application is deployed on the App Store. Following this phase, users start downloading and using the application. Their feedback and user experience over time, submitted to us via a dedicated customer support email channel or the CenceMe discussion board [2], trigger the application fixing and refinement process, in order to meet users satisfaction and improve the application usability. In what follows, the lessons learnt from the iPhone CenceMe deployment are reported.

Information Disclosure. When leveraging an application distribution system such as the App Store to collect data to be used for research purposes, it is very important that the user downloading the application is fully informed about the nature of the application and the data being collected, as well as how the data is going to be handled. Full disclosure of such information is often required by universities IRB and disclaimers should be made clear in the application terms of service. Given the sensitive nature of the iPhone CenceMe data, i.e., inference labels from sensor data, our university IRB makes us add a different consent form following the terms of service page which explicitly mentions the purpose of the application and describes the nature of the data collected along with the use we are making of the data.

According to IRB, this extra step is needed because people do not often read terms of service notes carefully, so a second dedicated disclosure form is required. Of course, by requiring a consent form through the involvement of IRB as often needed when carrying out research involving human subjects, the cycle of an application deployment becomes much longer. The IRB might take months to approve a certain research project, and even so several iterative steps may be needed in order to meet the IRB requirements. This implies long cycles before an application can be released. This extra time should be taken into consideration by researchers that want to carry out research at large scale through application distribution systems. The second implication of adding an explicit consent form in the application is that users might opt out from using the application (as we verified with some of the iPhone CenceMe users). This is because people are not yet used to downloading research applications from a commercial platform such as the App Store and they do not often understand the purpose of the research. As a consequence, the pool of users participating in the research might grow slowly.

Monetary and Time Costs. Moving research outside the lab for large scale deployments through the App Store has also monetary and time related costs. Bursty incoming user data, along with the necessity to rely on robust and reliable backend servers, most likely demand the support of cloud computing services [1]. In this way the researchers maintenance job is greatly alleviated since existing cloud services guarantee reliability and the ability to rapidly scale to more resources if needed. The flip side is that researchers have to be ready to sustain the subscription cost.

It is also to be taken into account the time overhead needed to adapt the application to new phone OS releases (which often carry API changes) in order to make the application able to transition through different versions of the software seamlessly. Without this support, users would not be guaranteed a smooth usage of the application which could potentially be dropped with severe impacts on the research outcome. Users might also ask questions and need to be guided through the use of the application. Researchers need to be ready to devote some of their time to customer service support. A prompt response from an application developer gives strong feelings about the solidity of the application and the people supporting it.

Software Robustness. Software robustness and clean user interface (UI) design may be a foregone conclusion. However, the effects of poor software design (which implies little robustness of the application) and poor UI layouts should not be underestimated. People downloading an application from any distribution system expect the software to be robust, simple to use, with easy-to-navigate UI. If any of these requirements are not met, users might loose confidence in the application and not use it anymore. As researchers, we might not have the UI design skills often required to make an application attractive. It is then important to collect feedback from domain experts that can guide the researcher to a proper design of the UI. We learnt about this issue after a couple of iterations of the iPhone CenceMe client. We modified the UI design and the different navigation views by taking into account users feedback and our own experience in using the app.

By dealing with software that needs to run on mobile phones, researchers have to pay great attention to the impact the application might have on the phone performance itself. Phone manufactures often guarantee that the user experience with the phone is not degraded when third party apps are running. Hence, resources are reclaimed, namely RAM and CPU, when the phone OS assesses that there is a need for it. Researchers have to make sure that the application does not take too many CPU cycles and/or occupy too much RAM, otherwise the application might be shut down unexpectedly. This is a particularly important aspect to be considered for applications designed to run in the background. Researchers that want to deploy applications at large scale have to be ready to write code at near-production level, in order to maximize the application usability and robustness.

Although testing the application in the lab might not let you discover all the possible glitches in the code, extensive testing phases are required before submitting an application to the distribution system. This is important in order to minimize the likelihood that users will encounter problems with an application and to reduce the chances that an application is rejected during the screening process; for example in the case of the Apple App Store. It should be noted that Android Market does not screen applications making it more attractive in the case of some applications. One of the CenceMe releases did not pass the screening phase because of a debugging flag was mistakenly left in the code causing the application to crash. As a result of this silly mistake the application was pushed to the back of the application screening process queue by Apple delaying the new release of CenceMe by several weeks.

Hardware Incompatibilities. New phone models or the evolution of existing models could present hardware differences that could impact the application performance. We experienced this issue during the iPhone 2G to 3G generation transition phase, where the former mounts a different microphone than the latter. We started noticing a performance drop of the audio classifier when the same application was running on a 3G phone. The drop was caused by the fact the audio classifier for conversation detection was trained using audio samples mainly recorded with iPhones 2G. Since the frequency response of the iPhone 3G microphone is different from the 2G model, the classifier trained with 2G audio was not able to infer accurately 3G audio. For a large scale application developer it is then important to realize these differences in time to limit misbehaviors when people replace their devices.

User Incentives. In order for users to use a research application they have to have an incentive and enjoy it when the application is active on their phone. If there is no or little return to them, the application might be used rarely with scarce benefit to the research outcome. In order to engage users we include a feature in the iPhone CenceMe client named *IconMe*. IconMe allows a user to select an icon that better represents their status, mood, and surroundings and associate a 140 character message to it. Such a message is shared with the CenceMe friends and posted on the personal Facebook, MySpace, and Twitter account according to the user's preferences. We found this microblogging service an effective way to stimulate the use of iPhone CenceMe.

User Reviews. Users reviews through blog posts or on the application distribution review system itself can be brutal. And the impact of such reviews may be negative for the application reputation. In order to limit the likelihood people report something not pleasant about the application, which might discourage others from downloading it, the testing phase preceding the deployment has to be extensive as pointed out above. For example, at the very beginning of the App Store launch we thoroughly tested iPhone CenceMe on legacy iPhones, i.e., not jailbroken – which have modified software to enable deeper interaction with the platform than what possible with legacy phones. It never occurred to us that jailbroken phones might induce unexpected behaviors when running our application. Some App Store reviews from iPhone CenceMe users were reporting unexpected crashes when using the application. After some investigation, we realized that the issue was present only on jailbroken phones (the reviews were in fact from jailbroken phone users) and caused by a routine accessing the sqlite database. It is again very important to point out that across-platforms tests are needed in order to make sure the application is robust. This is a practice that is not usually required for limited lab setting experiments, but necessary for large scale deployments and is again a factor that impacts the duration of the research.

Software Limitation. Although sensors are being included in smartphones, the APIs to access them and their behavior is not entirely designed to support mobile sensing applications. An example is the accelerometer APIs. Phones operated by Android OS and the recent new iPhone OS4 support multitasking and give the possibility to run applications in the background. Activity monitoring applications require continuous streams of accelerometer data. However, since the accelerometer on modern smartphones is mainly intended to drive the user interface (e.g., move from portrait to landscape mode) the manufacturers opted not for delivering accelerometer data to the application anymore when the app is pushed to the background. Although this is a sound design choice from a UI standpoint because an application in the background does not need an active UI, it is not desirable for a continuous sensing application. Such a limitation has to be taken into consideration.

Lack of Ground Truth. As soon as a mobile sensing application is being used and inferred labels start converging to the backend, the question is: how reliable that label is? Is the event being inferred actually occurring? We do not have ground truth evidence when the application operates in the wild. One way to increase the trustworthiness of the data is to randomly prompt the user to provide a label for the current inferred event. Even if this procedure is sparse in time, it might allow us to collect significant statistical evidence in the long run. The other option is to design and exploit novel techniques that mine peoples multimedia content, (e.g., microblog posts/tweets, video, photos, audio clips) as a means of seeking correlation between the semantic of keywords/characteristic features of the multimedia content and the actual activity. We are currently investigating both the possibilities.

RELATED WORK

Researchers have started leveraging commercial application distribution systems to carry out global scale research [9, 11, 8, 5] while identifying challenges [7]. The authors of [8] analyze the public available data from the Brightkite mobile social network to study the behavior of users. Results from the study of a large deployment through the App Store of a game application are reported in [9], while [11] discusses the findings from a global scale study of a HCI technology deployed on the App Store. The authors of [6] and [5] present the results of medium and large scale experiments to investigate smartphone usage patterns, respectively.

CONCLUSION

We presented an overview of the design of the CenceMe application and discussed some of the lessons learnt from our deployment through the Apple App Store. We gained a number of important insights using new application delivery channels presented by app stores and supporting a large number of users over a long period of time for what is essentially a research app and not a commercial app - a sort of Trojan horse of sorts, a research project masquerading as a phone application. Many important questions remain in this new era. How do we collect and validate our research data when we have limited control over users and lack real ground truth? How do we make sure we have a good cross section of users to validate our study? Will app stores continue to allow academic researchers to use their deliver channels to do research? If a research app becomes wildly popular how do small academic labs respond to that in terms of financing cloud infrastructure and supporting potentially 100s of thousands of users? It is clear that the new environment represents a fast way to try out new ideas in the market place driving more and more innovation. In essence, the app stores are the digital equivalent of the petri dish - we can germinate new ideas and watch them grow or fade in the real world, with real users, distributed across the world. This is a very exciting departure from how we did research before app stores.

REFERENCES

1. Amazon Web Services (AWS). `http://aws.amazon.com`.
2. CenceMe Discussion Board. `http://getsatisfaction.com/cenceme`.
3. LIBSVM. `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.
4. PEIR. `http://peir.cens.ucla.edu`.
5. Earl Oliver. The Challenges in Large-Scale Smartphone User Studies. In *HotPlanet 2010*. June 15, 2010, San Francisco, USA, 2010.
6. Falaki, H. et al. Diversity in Smartphone Usage. In *MobiSys 2010*. June 16–18, 2010, San Francisco, USA, 2010.
7. Henderson, T. and Abdesslem, F.B. Scaling Measurement Experiments to Planet-Scale: Ethical, Regulatory and Cultural Considerations. In *HotPlanet 2009*. Krakow, Poland, 2009.
8. Li, N. and Chen, G. Analysis of a Location-based Social Network. In *SIN 2009*. Vancouver, Canada, August, 2009.
9. McMillan, D. et al. Further into the Wild: Running Worldwide Trials of Mobile Systems. In *Pervasive 2010*, 2010.
10. Miluzzo, E. et al. Sensing Meets Mobile Social Networks: The Design, Implementation and Evaluation of the CenceMe Application. In *SenSys 2008*. Raleigh, NC, USA, Nov. 5–7, 2008.
11. Zhai, S. et al. ShapeWriter on the iPhone From the Laboratory to the Real World. In *CHI 2009*, 2009.