Liberating Mobile Phones from their Primary Use Case

Ivan Beschastnikh[1] Yuan Zhang[2] Zhengping Qian[3] Lidong Zhou[3]
[1]University of Washington [2]Tsinghua University [3]Microsoft Research Asia

Abstract

A technical limitation of today's mobile phones is that they are designed for a narrow set of primary use cases. This is one of the reasons why the majority of mobile phones are disposed of or relegated to the drawer even though they are fully functional.

In this paper we argue that future mobile devices should be designed to support use cases beyond their primary purpose. We motivate and explore the implications of two design principles that can help us move towards this vision – *loose association* and *device composition*. We discuss an assortment of mobile applications that are enabled by these principles and present our early experiences in applying these principles in the context of a scheduled sensing system deployed on used phones.

1. Introduction

Mobile phones are essential to our day to day activities – over a quarter of the world's population has a mobile phone [11]. Unfortunately, mobile phones have an average use lifetime of just 17.7 months [14]. The fact that most replaced phones are usually fully functional but rarely reused [7] underscores an important *design* limitations – mobile phones are designed for a narrow set of primary use cases.

We believe that the tyranny of the primary use case (1) hampers mobile smart phones, which are highly versatile networked sensing, computing, and storage platforms; and (2) promotes short mobile phone lifetimes, which adversely impacts the environment.

Exposing smart phone versatility. Almost all of today's mobile applications either require or assume that the host phone is the user's primary device. However, constant innovation in sensing, storage, and processing capabilities enable a potentially disruptive model of autonomous operation. This model encourages applications in domains that are traditionally considered off-limits to off-the-shelf smart phones.

For example, just one smart phone is sufficient to develop a baby monitor on par with custom solutions. The parent's primary mobile device can be alerted with an SMS message or an email in case the baby is detected as missing from its crib, or if it screams for too long. Section 4 presents more application examples.

Environmental concerns. Just 3% of the US population recycles mobile phones [11]. In a survey of undergraduate students with an average age of 19.7 years, Hanks et al. found that over 30% of the students owned 4-8 mobile phones in their lives [7] – a tremendous replacement rate.

One way to experiment with many of the secondary mobile phone use cases we envision in this paper is to build systems on top of used mobile phones. The fast replacement cycle of mobile phones may be seen as an opportunity since used mobile phones typically include sensors and capabilities that characterize the latest models. Used mobile phones are also fully functional as replacement is driven not by wear-and-tear or malfunction but by fashion trends and contract models [7, 8].

We believe that systems researchers are well placed to re-design the software stack of mobile devices to support alternative secondary uses. In this paper we outline two design principles that help with this. We also present a variety of mobile applications that are enabled by these principles and discuss our early experiences in applying these principles in the context of a scheduled sensing system deployed on used phones.

2. Relaxing the user – device association

Smart phones have proven to be successful at fostering innovation in the domain of mobile applications. However, almost all of today's mobile applications either require or assume that the host phone is the user's primary device.

Mobile phone design is highly influenced by the tight association between a device and its user. For example, this association impacts the software stack – mobile phones generally have no server functionality to host content or services, and are designed to support user-facing applications that are highly interactive. As well, most devices have a physical form factor that precludes non-handheld use cases – it is usually impossible to stand a phone vertically on a flat surface without extra support. The tight user – device association makes it challenging to leverage mobile phones in scenarios that preclude such tight association.

The notion of *loose association* between a user and a mobile device relaxes the interactivity constraint. Because modern phones have plenty of storage, processing, and sensing resources, the phone can continue to serve a useful function to the user even if the user never directly interacts with the device. This disassociation may be complete and the phone may only carry out tasks on a user's behalf through other devices. It may also take on a relaxed form in which the user infrequently interacts with the phone.

Phones in particular, are in an ideal position to support loose association designs. Unlike PCs, mobile phones are small, unobtrusive, use little power, and have rich sensing capabilities.

2.1 Implications of Loose Association

Users interact with loosely associated mobile phones infrequently or not at all. Because these devices receive little user attention we have to reconsider the hardware and software requirements of the mobile phone, along with important aspects such as mobility and power constraints.

Device is stationary or rarely mobile. The user will not burden themselves with actively carrying multiple phones around with them. A dis-associated phone is expected to be stationary unless it is located in some mobile space, for example, a vehicle. This lack of mobility means that device placement has to be carefully considered.

Without mobility certain device sensors will report trivially constant readings – e.g., GPS coordinates or magnetic orientation. For other sensors, like the camera, lack of mobility can be leveraged to perform delta computation on sensor reading to conserve storage and bandwidth.

Device is always powered, and always on. The user cannot be expected to periodically re-charge the dis-associated device or bother to frequently turn the phone on/off. The phone will therefore quickly run out of power unless it is always plugged in. To be useful the phone must also be always kept on.

This implication greatly impacts the phone software and hardware. For example, power management becomes less critical and the phone does not need to power off its radios, or optimize for power by sleeping or scaling CPU frequency.

The validity of this implication depends on the amount of power the user will dedicate to the deployed phones and the availability of an easily accessible and uninterrupted power source. On the other hand, most mobile phones are power-efficient and have different modes of operation, which can be leveraged to save power. With conservative use, a mobile phone's battery can last for many hours. This enables scenarios in which the phone is charged with solar energy during daytime and expends the stored energy at night.

Device rarely receives direct user input. Systems composed of loosely associated phones may operate autonomously and fade into the background. As an enhancement, we think that users may also want to interact with such systems indirectly, through the user's primary mobile phone or a PC.

This has repercussions for the software stack. In contrast to popular mobile operating systems such as Windows Mobile, we believe that loosely associated devices can be fully functional with radically simpler operating systems. The operating system can exclude many complex drivers, such as those for handling user input and using the display, and can be designed to be more sensor-driven. We think that an OS similar to that of TinyOS [4] is more appropriate for loosely associated devices.

However, the disassociation does not have to be complete. The association may be flexibly adjusted depending on the application scenario and capabilities of the phones in the system. For example, a phone's display can be used as a passive display screen, or the device may support only simple interaction modalities such as selection of a textual choice from a short menu.

Device must be networked. The user will not spend time to synchronize the phone with a computer or other devices. The phone must be able to communicate with other phones to relay sensor data, receive commands, etc. This means that the phone must expose a semantically rich interface to allow remote control of functionality that has previously been only accessible through the user interface.

Another impact of relaxed association on networking is a change in how the device uses the network. Because users tend to use mobile phones as gateways to information on the web, most phones in active use have a download to upload ratio greater than 1. However, a loosely associated phone that acts primarily as a sensor will have a ratio value of less than 1.

Automated application deployment. A challenge for systems composed of loosely associated phones is software deployment. The user cannot be expected to attend to each phone whenever they would like to install a new application on the phones. For deployment users can leverage their PC or a primary mobile phone as a control point from which to communicate and distribute commands to the loosely associated devices.

3. Device Composition

The networking requirement due to loose association can be leveraged to compose groups of phones into networks and into virtual phones. Here we discuss these two composition types and their implications for system design.

3.1 Composing Networks of Devices

Modern mobile phones are typically equipped with multiple network interfaces. This naturally motivates a composition where multiple phones are networked together. This composition raises traditional networking challenges including naming, discovery, and routing. Many of these issues have been thoroughly researched in the space of ad-hoc wireless networks, and sensor networks. Networking phones also implicates distributed systems issues as the phones must coordinate to implement application logic.

Loose association limits mobility, eases power constraints, and implies uninterrupted connectivity. Moreover, users are not expected to interact with the phones directly, which dramatically alters the network workload. We believe that all of these factors present a unique combination of challenges for networking loosely associated mobile phones. Investigating these challenges in detail is future work.

3.2 Implications of Composing Device Networks

Internet connectivity via routing. If networked, a phone can reach the Internet by routing to an Internet-enabled phone. This provides Internet connectivity to phones that were originally not intended to access the Internet.

Programmability and coordination. A key challenge to reaping the benefits of networked phones is devising abstractions and services that make it easy to program and coordinate groups of phones.

3.3 Composing Virtual Devices

Multiple physical phones may also be composed into a virtual phone, whose external behavior is indistinguishable from that of a single device. Virtual phones enable a number of beneficial scenarios in which physical phones complement each other's capabilities or cooperate in some way. For example, a phone with a microphone can be paired with another phone that has a camera to create a virtual phone that can record videos.

This composition may be achieved in a few ways. If the phones can be co-located then a physical connection between them, such as a USB cord/hub, offers advantages like high throughput and good reliability. The physical phones may also be connected using a wireless channel, however, this composition can be detected and disrupted by nearby devices.

A virtual device is a tightly integrated distributed system. One arrangement is to elect one of the physical device as the leader, and have the other devices function as replicas. A distributed consensus algorithm, such as Paxos [10], can be used for this. During leader failures a replica can be elected to become the new leader. Externally, all communication with the virtual device would proceed via the leader.

3.4 Implications of Composing Virtual Devices

Improved reliability. Composing multiple phones into a virtual phone also provides higher reliability through redundancy. For example, the phones may improve device state reliability through replication. Additionally, if the phones support the same sensing capabilities (e.g., all phones have a microphone) then the virtual phone can continue to provide sensor readings even as physical phones or their sensors fail. Multiple sensors can also be used for higher signal fidelity.

Improved device resources. Phones can be composed to create virtual phones with higher resources at their disposal. For example, a phone with a large storage capacity (e.g., an iPhone) can be used for storage in a virtual device which also includes different phones with valuable sensing capabilities but with little or no storage capacity for sensor readings.

4. Enabled Applications

Mobile phones can be leveraged for a variety of alternative use cases in different settings. Here we present applications in three settings – home, office, and in a powered mobile environment, like a car. Although some of these applications have commercially available *custom* solutions, the approach of reusing mobile phones with the design principles of loose association and device composition provides a *generic* means to enable these applications with commodity smart phones as components. We believe that mobile phones can catalyze innovation that is un-hindered by specialized hardware or obscure protocols.

4.1 In the Home

Specialized surveillance equipment is usually necessary for a home monitoring system. Such systems are expensive, highly customized, and are difficult to extend or modify. Mobile phones with rich sensing capabilities can be leveraged to provide similar functionality. A phone with an accelerometer that is attached to a door can track it opening/closing. A phone with a light sensor can detect when someone is up late watching TV. The ubiquity of phones with cameras and microphones enables recording of interesting events.

Mobile phones can also be used to create advanced intercoms that stream video and voice between devices (e.g., for communication between the front door and the backyard). Similarly, a system of phones can implement a smart baby monitor. The parent's primary mobile device can be alerted with an SMS message or an email in the case that the baby is detected as missing from its crib, or if it screams for too long.

Mobile phones can also enable localized but asynchronous communication in busy families. For example, messages in the form of voice, video, or chat can be stored on a device deployed on a kitchen counter and created/viewed from the users' primary mobile devices.

Unlike customized solution, phones can be adapted to numerous applications. Additionally, their connectivity can be used to yield sophisticated personalization. For example, a system can access the family's web-based calendar and make inferences about events in the home.

4.2 At the Office

Mobile phones can be used to monitor human activity at the office. For example, conference rooms at Microsoft Research can be reserved ahead of time by the researcher. A deployment of mobile phones in a conference room enables numerous applications [5]. One problem we observed is that sometimes reservations are not used – all rooms seem unavailable, yet some are empty. A phone could track usage and alert the researcher to relinquish the reservation when the room is not used at the time of the meeting. Phones could also track who attended the meeting by detecting and resolving the attendees' ID cards, or bluetooth-enabled mobile devices. Everyone at the meeting could then receive an email with a list of attendees.

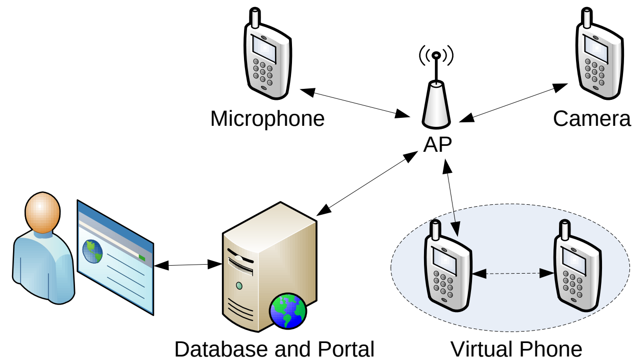

Figure 1. An overview of Sensitivity's design.

4.3 In a Mobile Environment

Loosely associated mobile phones do not have to be stationary. They can be attached to vehicles, such as a car or a bus. For example, the phone could be used to record the vehicle's location whenever it is mobile. When the vehicle returns to the garage, the phone can upload the trip information to the network. This may then be used to compute more optimal routes personalized by actual routes taken and the associated travel time. The trip information can also be used to tag pictures/videos uploaded by the user from their digital camera with location information – by associating the time the picture/video was taken with the GPS location of the vehicle at that time. The user's primary phone can accommodate some of these scenarios, but it is associated with the user and not the vehicle, therefore it must do more work to deduce the user's mobile context and associate it with the correct vehicle.

5. Experiences with Sensitivity

We designed and implemented *Sensitivity*, a system for scheduled sensing. Sensitivity is composed of a set of used heterogeneous mobile phones and a PC computer. It allows users to group phones into virtual phones and to schedule sensing tasks across phones. Users may then access the collected sensor data directly, or application developers may build user-facing applications that leverage the data collected by Sensitivity. We note that the technology to enable phones to perform remote sensing already exists [12]. Our intention with Sensitivity is to explore design principles, such as loose association and device composition, that enable systems that use mobile phones in new ways.

5.1 Design

In Sensitivity, the mobile phones are completely disassociated from the user. Once the software is uploaded to the phones (currently, this is done manually), the phones operate without any user intervention.

Initially, the phones advertise their sensor stack to the PC. Then, the user accesses a web-based portal hosted by the PC to specify a cron-formatted schedule for some *device.sensor* known to the PC. The PC distributes the appropriate schedules to each device and the phones sense according to the last schedule they received from the PC, pushing the collected sensor data to the PC, which stores this data in a database. The user can then access, query, and download the collected sensor data from the web-based portal. Figure 1 illustrates our design.

Simplicity. We made a simplifying fate-sharing assumption that it is reasonable for our system to fail in case of the PC failure. This enabled us to centralize all control at the PC, which greatly simplified our mobile device code and the system's communication topology to a star. Also, we composed our phones into a network by reusing the existing 802.11 infrastructure.

Modularity. Inspired by little languages [3] each device uses small specialized sensor collector programs to collect and process individual sensor readings (e.g., read from a microphone, or take a picture). A sensing scheduler program communicates with the PC and directs sensor collectors to perform their task according to the current schedule.

Interfaces. Our design leverages web-interfaces to coordinate all communication between the PC and the devices. Additionally, we use the file system to communicate sensor readings between the sensing scheduler and the sensor collector programs on each device. Finally, the user can access the web-portal from any device that can reach the PC. For the user this makes the choice of portal host less important.

Virtual Phones. As our Symbian phone supports only Bluetooth, we paired it with another phone that relayed communication between the Symbian phone and the PC through an 802.11 connection. This virtual composition greatly simplified development overhead in networking these devices. Additionally we composed a virtual device to pair two phones with complementing sensors – one phone with a microphone, and another phone with a camera. The new virtual device can be scheduled to take movies with audio.

5.2 Implementation and Deployment

We deployed Sensitivity in an office setting. Our deployment includes used mobile phones running Symbian, WM6, and WM5. Each device is plugged into an outlet and is positioned to capture a variety of sensor data. We use a typical desktop PC as our server.

We leave evaluation of Sensitivity to future work. Our experience with applying loose association and device composition principles is preliminary, but we have already found these principles to be useful guides in considering system design alternatives.

6. Discussion

One way to experiment with many of the secondary mobile phone use cases we envision is to build systems on top of used mobile phones. These devices are cheap, plentiful, and pose a major environmental concern. However, they have a number of drawbacks – used phones are not as reliable as new devices and may have fewer resources (e.g., less processing power); as well, they may not support the latest software frameworks and protocol stacks. Notwithstanding these challenges, we believe that used devices are great candidates for experiments that require many cheap devices.

A viable path to deployment and commercialization of mobile phones that target secondary use cases is less clear. Existing devices have developed into platforms that are capable of much more than effective support for the primary use case. We believe that it is simply a matter of time before this untapped potential is leveraged more fully by mobile application developers, who will in turn influence the design of future mobile devices.

7. Related Work

Although the focus of this paper is on mobile phones, much of the discussion naturally extends to other devices, such as MP3 players and digital cameras. For instance, iPods have significant storage capacity and can be used for storage.

Work by Huang et al. [8] points out the tight degree of association between mobile devices and users as one reason to explain the high discard rate of mobile devices. This work informs the idea of loose association. In prior work Jain and Wullert II outlined important challenges for designing environmentally sound pervasive systems [9]. Loose associated and device composition, the two principles introduced in this paper, can be used to mitigate some of their documented challenges.

Mobile phones provide essential infrastructure in developing regions [13]. Systems designed to enable secondary uses of mobile phones can benefit from such deployment experiences, and can also inform the design of systems for use in developing regions.

An important design consideration for supporting multiple use cases is modularity. Unfortunately today's mobile phones are tightly integrated hardware and software platforms. A more modular design could significantly benefit alternative uses [2].

Composing mobile devices is especially difficult because of limited interoperability. Recombinant Computing [6] explored runtime compatibility between mobile devices over many years. The lessons and experiences of these efforts, particularly the design of "meta-interfaces," are directly applicable to networking multiple mobile devices to support alternative use cases.

Numerous related systems work explores how rich computing environments may be enabled with mobile devices. For example, in the Dynamic Composable Computing [15] proposal devices are composed on demand to enable application scenarios that afford mobile users a highly flexible computing experience. In cyber foraging [1] mobile devices opportunistically use resources as they move about. In this paper we focus on alternative uses cases for mobile phones and propose design principles to guide systems supporting such uses cases.

8. Conclusion

Today's mobile phones are designed for a narrow set of primary use cases. This hampers mobile smart phones, which are highly versatile networked sensing, computing, and storage platforms; and promotes short mobile phone lifetimes, which adversely impacts the environment. We presented two design principles to support alternative secondary uses of mobile phones. These principles are motivated by the intuition that future mobile devices will possess sufficient flexibility to enable new application scenarios that depend on little to no user participation (loose association) and will have sophisticated inter-device communication capabilities (device composition).

References

[1] BALAN, R. K., GERGLE, D., SATYANARAYANAN, M., AND HERBSLEB, J. Simplifying cyber foraging for mobile devices. In *MobiSys '07* (New York, NY, USA, 2007), ACM, pp. 272–285.

[2] BARBA, E. Getting mod: a look at modularity in mobile systems. In *HotMobile '08* (New York, NY, USA, 2008), ACM, pp. 22–26.

[3] BENTLEY, J. Programming pearls: little languages. *Commun. ACM 29*, 8 (1986), 711–721.

[4] BERKELEY, U. Tinyos community forum —— an open-source os for the networked sensor regime. (2009). `http://www.tinyos.net`.

[5] CHEN, H., PERICH, F., CHAKRABORTY, D., FININ, T., AND JOSHI, A. Intelligent agents meet semantic web in a smart meeting room. In *AAMAS '04* (Washington, DC, USA, 2004), IEEE Computer Society, pp. 854–861.

[6] EDWARDS, W. K., NEWMAN, M. W., SEDIVY, J. Z., AND SMITH, T. F. Experiences with recombinant computing: Exploring ad hoc interoperability in evolving digital networks. *ACM Trans. Comput.-Hum. Interact. 16*, 1 (2009), 1–44.

[7] HANKS, K., ODOM, W., ROEDL, D., AND BLEVIS, E. Sustainable millennials: attitudes towards sustainability and the material effects of interactive technologies. In *CHI '08* (New York, NY, USA, 2008), ACM, pp. 333–342.

[8] HUANG, E. M., AND TRUONG, K. N. Breaking the disposable technology paradigm: opportunities for sustainable interaction design for mobile phones. In *CHI '08* (New York, NY, USA, 2008), ACM, pp. 323–332.

[9] JAIN, R., AND WULLERT, II, J. Challenges: environmental design for pervasive computing systems. In *MobiCom '02* (New York, NY, USA, 2002), ACM, pp. 263–270.

[10] LAMPORT, L. The Part-Time Parliament. *ACM Transactions on Computer Systems 16*, 2 (1998), 133–169.

[11] NOKIA. Global consumer survey reveals that majority of old mobile phones are lying in drawers at home and not being recycled. (2008). `http://www.nokia.com/press/press-releases/showpressrelease?newsid=1234291`.

[12] NOKIA. Nors - nokia remote sensing platform. (2009). `http://opensource.nokia.com/projects/nors/index.html`.

[13] PARIKH, T. S., AND LAZOWSKA, E. D. Designing an architecture for delivering mobile information services to the rural developing world. In *WWW '06* (New York, NY, USA, 2006), ACM, pp. 791–800.

[14] POWER, J., AND ASSOCIATES. U.s. wireless mobile phone evaluation study. (2008). `http://www.jdpower.com/corporate/news/releases/pressrelease.aspx?ID=2008059`.

[15] WANT, R., PERING, T., SUD, S., AND ROSARIO, B. Dynamic composable computing. In *HotMobile '08* (New York, NY, USA, 2008), ACM, pp. 17–21.