Tapping Your Phone's Context Continuum

David Chu[*], Aman Kansal[+], Jie Liu[+], Bodhi Priyantha[+]

[*]Microsoft Research Asia
No.49, Zhichun Rd
Beijing, China 100190

[+]Microsoft Research
One Microsoft Way
Redmond, WA 98052

The recent smartphone revolution is primarily a sensing and service revolution. A typical smartphone on the market today has more than eight sensors, such as microphone, camera, GPS, accelerometer, digital compass, gyro, light, and the touch screen. These sensors not only provide natural user interaction on the portable devices, but also offer exciting opportunities for pervasive sensing and context-aware pervasive computing.

Ostensibly sensor stimuli from the outside world should be guiding and informing much of our everyday activities. Pervasive computing implies continuity through time and space. Yet at the moment phones are achieving neither. Currently, meaningful phone interactions require a high degree of user initiative and can only take advantage of instantaneous context. That is when an application requires context input, it samples the sensors and reacts accordingly.

The Vision. We see opportunities for mobile phone service architectures to address the systems challenges of pervasive sensing in both time and space. We take the unorthodox view that every app's conversion of raw sensor data to *Application Data Units (ADUs)* be delegated to a *Sensor Processing Service (SPS)*. Traditionally applications start with a very low level interface to data: anonymous byte streams. We suggest that the right approach for mobile systems is a much higher level of data abstraction. We propose raising the bar all the way up to the ADU: the application's meaningful unit of data. For example, a transportation mode application may use ADUs indicating whether a user is waking, running, cycling, or driving, inferred from GPS and IMUs.

What does it mean for a SPS to manage ADUs? Bytes are universally unassuming, whereas ADUs are by definition highly application-specific. Rather than potentially support an incredibly diverse ADU interface directly, it could be possible to let applications to delegate their sensor sampling and processing functions to SPS. The service will manage the continuous execution of these functions, and return ADUs to each application as appropriate.

Potential Major Advances as a result of the Vision. The challenges an SPS might tackle include (1) scheduling and energy management of continuous sensing and processing, (2) cross-system optimizations regarding ADU generation, and (3) sensor data sharing amongst users.

First, current smartphone are extremely cautious with the types of background processing available due to their unpredictability. By passing responsibility for ADU generation to SPS, applications may be relieved from worrying about managing energy concerns. At the same time, the data processing service might efficiently service low energy situations by paring back ADU generation in a principled way when resources are limited (e.g. when foreground apps are activated) that ultimately maintains the perception each app has an uninterrupted ADU stream. Our previous work a-Loc[i] and Naku[ii] may provide a starting point here. a-Loc focused on trading off accuracy and energy-efficiency for continuous location streams.

Naku showed that even complex sensor processing is in fact highly predictable with the proper profiling, and amenable to graceful quality degradation when resources become tight.

Second, a unified processing service could better plan cross-systems optimizations. These optimizations could come on two fronts. First, as more and more applications consume ADU context, SPS could mediate results sharing of sensor sampling and processing stages amongst apps. For example, a single application's interest in geolocation may not warrant an expensive but accurate GPS request, but multiple applications that can all share in the benefits might. ADUs as well as intermediate results (such as FFTs performed on an audio signal) may be candidates for sharing. The SPS's challenge is to identify and manage these sharing opportunities. Second, mobile platforms are increasing in hardware diversity: coprocessors and gpgpus may soon emerge on some phones. App developers are frequently stymied by phone platform variations. Systems support map sensor sampling and processing to the appropriate hardware is needed. Our work on LittleRock[iii] showed that main processor offloading to a subsidiary microprocessor could substantially reduce the overall system power needs for continuous sensing. The issue of how specialized processors can best be engaged for arbitrary ADU processing warrants investigation.

Finally, enforcing secure sensor data sharing policies is important. In a world where sensing is truly pervasive through space, phones ought to be able to collaborate to achieve wide- or local-area sensing goals. However, users may very well be wary of sharing raw sensor data due to the embedded personal information. Sharing works best when users can fine-tune their shared data's degree of abstraction. An intermediate data distribution platform should provide the mechanisms for appropriate data sharing. For example, by working with the ADUs directly, the distribution platform can limit distribution of raw data, and only share only selected ADUs.

The idea of ADUs as the narrow interface may at first seem counterintuitive. Yet in fact, raising the abstraction may be well in line with current trends. Geolocation is often used in place of GPS, wifi and cellular beaconing. Touch screen input is often abstracted as pinch and zoom gestures. These point solutions may only hint at the beginning. We are excited by continuous sensor processing because it offers intriguing possibilities for making context truly pervasive.

Background and Experience of Participants. David Chu has worked on sensor programming languages and mobile data processing. Aman Kansal has worked on fine-grained energy monitoring for mobile, and adaptive location processing for mobile. Bodhi Priyantha has worked extensively on low-power systems including phone co-processors for continuous sensing. Jie Liu has worked in many areas of embedded sensing systems, including continuous phone sensing. See our papers below for more info.

---

[i] Energy-accuracy trade-off for continuous mobile device location. Kaisen Lin, Aman Kansal, Dimitrious Lymberopoulous, Feng Zhao. MobiSys 2010.

[ii] Naku: Balancing Energy, Latency and Accuracy for Mobile Classifiers. David Chu, Tsung-te Lai, Nicholas Lane, Cong Pang, Feng Zhao. Preliminary report at: http://www.bawakayi.com/davidchu/2010/06/13/continuous-sensing-and-classification-on-mobile-phones/

[iii] Continuous Sensing on Mobile Phones with LittleRock. Bodhi Priyantha, Dimitrios Lymberopolous, Jie Liu. Poster. MCS 2010.