MobSens: Making Smart Phones Smarter

Four mobile sensing applications that work on off-the-shelf mobile phones contain elements of health, social, and environmental sensing at both individual and community levels.

Eiman Kanjo, Jean Bacon, and Peter Landshoff
University of Cambridge

David Roberts
Nokia

Today's mobile phones are smarter than ever: they now take and process pictures and videos, issue messages and email, access the Web, allow games on demand, and play music. More people around the world take their phones everywhere they go, using them in a variety of environments and situations to perform a whole range of different tasks. In India, for example, more people access the Internet from their phones than from a PC, a scenario that will certainly play out across the globe in the years to come.[1]

Most mobile phones include a variety of sensing components. By expanding this capability, we can derive some interesting sensing modalities—for example, scrutinizing local environments to detect and reduce pollution or using medical applications to tackle other problems on a societal scale.

In this article, we discuss experiences and lessons learned from deploying four mobile sensing applications on off-the-shelf mobile phones within a recreational framework called MobSens that contains elements of health, social, and environmental sensing at both individual and community levels. We describe the main components of our applications, which facilitate logging and external communications. We also outline the challenges faced when building and testing these applications and describe our strategies for overcoming them.

MobSens Prototypes

Mobile sensing—also known as "participatory sensing,"[2] "urban sensing,"[3] or "participatory urbanism"[4]—enables data collection from large numbers of people in ways that weren't previously possible. Using mobile phones has several advantages over unattended wireless sensor networks for environmental sensing applications:

- Mobile phones can provide coverage where static sensors are hard to deploy and maintain, and large numbers of cell phones already exist around the world, providing the physical sensing infrastructure.
- Deploying the sensing hardware and providing it with network and power requires significant effort in other sensor networking systems. The availability of more powerful operating systems and the transfer of standardized programming languages on ever-smaller computing platforms have spurred the recent development of software applications for mobile computers, including Symbian (www.symbian.com), Google Android (http://code.google.com/android), Microsoft Mobile (www.microsoft.com/windowsmobile), and iPhone (www.apple.com/iphone).
- Such systems can benefit from local communities as the driving element for environmental sensing. This approach, sometimes referred to as "citizen science," uses mobile sensor technology to help individuals personally collect, share, compare, and participate

Published by the IEEE CS ■ 1536-1268/09/$26.00 © 2009 IEEE

Related Work in Mobile Sensing

Research progress in wireless and sensor networking in the past decade has been astounding; recent developments in sensor networks in which the nodes are mobile and carried by people or vehicles have also emerged.[1,2] In particular, industry leaders are making some headway into changing the mobile sensing paradigm. For example, SensorPlanet (www.sensorplanet.org) is a Nokia-initiated global research framework for mobile device-centric wireless sensor networks that views mobile devices as both gateways to mesh sensor networks and as sensor nodes themselves.[3,4]

Various projects[5–7] have shown how the integration of sensors and positioning technologies used in conjunction with mobile computing devices can support data collection for environmental applications, without the overheads and complexities of wireless sensor networks. Some of these projects have considered the embedding of direct awareness in mobile devices, which is boosted by the rapid advance in sensor technology.

MobSens builds on a large body of related projects that use mobile phones as sensing devices. The MetroSense project, for example, outlines a "people centric" approach to mobile phone sensing that includes several deployments with bicycles.[8]

The Center for Embedded Network Sensing has a research initiative called "participatory sensing" that's developing the infrastructure and tools to let individuals and groups initiate their own public "campaigns" for others to participate in by using networked mobile devices.[9] The MyExperience tool is a mobile software application for *in situ* data collection that supports the study of human behavior and the evolution of mobile computing technologies.[10] MyExperience can also record a wide range of data, including information from sensors, images, video, audio, and user surveys.

The Metro project allows mobility-enabled interactions between human-carried mobile sensors via static sensors embedded in the civic infrastructure and wireless access nodes that provide a gateway to the Internet.[11] Finally, Nokia has introduced a new phone concept called "Nokia Eco Sensor" (www.nokia.com/A4707477). Its wearable sensor unit will house several optional sensors to monitor environment, health, and local weather conditions on a dedicated mobile phone.

REFERENCES

1. B. Hull et al., "CarTel: A Distributed Mobile Sensor Computing System," *Proc. 4th ACM Conf. Embedded Network Sensor Systems* (SenSys 06), ACM, 2006; http://db.csail.mit.edu/pubs/paper.pdf.
2. S.B. Eisenman et al., "MetroSense Project: People-Centric Sensing at Scale," *Proc. Workshop World-Sensor-Web* (WSW), ACM Press, 2006; http://metrosense.cs.dartmouth.edu/.
3. V.H. Tuulos, J. Scheible, and H. Nyholm, "Combining Web, Mobile Phones and Public Displays in Large-Scale: Manhattan Story Mashup," *Proc. 5th Int'l Conf. Pervasive Computing*, LNCS 4480, Springer, 2007, pp. 37–54.
4. T. Abdelzaher et al., "Mobiscopes for Human Spaces," *IEEE Pervasive Computing*, vol. 6, no. 2, 2007, pp. 20–29.
5. E. Kanjo et al., "MobGeoSen: Facilitating Personal Geosensor Data Collection and Visualization Using Mobile Phones," *Personal and Ubiquitous Computing*, vol. 12, no. 8, 2008, pp. 599–607; http://portal.acm.org/citation.cfm?id=1416970.
6. A. Steed and R. Milton, "Using Tracked Mobile Sensors to Make Maps of Environmental Effects," *Personal and Ubiquitous Computing*, vol. 12, no. 4, 2008, pp. 331–342.
7. P. Adamczyk et al., "Urban Computing and Devices," *IEEE Distributed Systems Online*, 2007; http://doi.ieeecomputersociety.org/10.1109/MDSO.2007.46.
8. A. Campbell et al., "PeopleCentric Urban Sensing," *Proc. 2nd ACM/IEEE Ann. Int'l Wireless Internet Conf.* (WICON 06), ACM Press, 2006; http://portal.acm.org/citation.cfm?id=1234179.
9. J. Burke et al., "Participatory Sensing," *Proc. World Sensor Web Workshop*, ACM Press, 2006; www.sensorplanet.org/wsw2006/6_Burke_wsw06_ucla_final.pdf.
10. J. Froehlich et al., "MyExperience: A System for In Situ Tracing and Capturing of User Feedback on Mobile Phones," *Proc. MobiSys 2007*, ACM Press, 2007; http://portal.acm.org/citation.cfm?id=1247670.
11. A.T. Campbell et al., "The Rise of People-Centric Sensing," *IEEE Internet Computing*, vol. 12, no. 4, 2008, pp. 30–39.

in interpreting the personal measurements of their daily lives.

- Mobile phones can directly pick up sensor data instead of having to send that data across an entire sensor network.
- Mobile phones can acquire, process, store, and transfer contextual data (such as photos and messages), which can be coupled with sensor data.

Essentially, our mobile sensing project aims to equip average citizens with mobile phone applications and tools that let them acquire quantitative and qualitative environment information directly from their surroundings. In the long run, this increased information could also promote "green" options not just in city streets but also in other aspects of daily life.

A variety of prototypes have emerged from our research to help a mobile

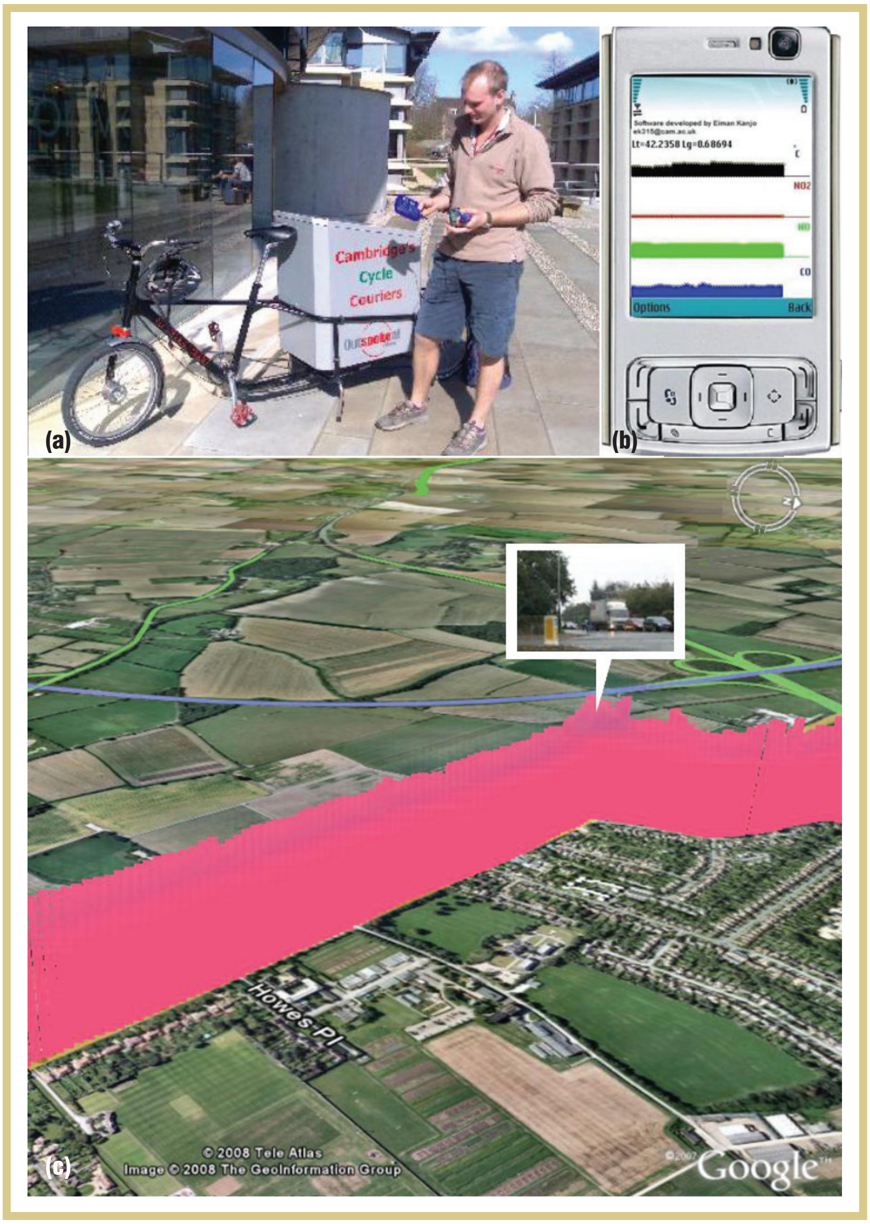


Figure 1. PollutionSpy. (a) One of our cycling couriers with a data logger and the N95 phone ready to collect pollution data. (b) Screenshot of PollutionSpy software. (c) A 3D track of CO data near a busy junction overlaid (in real time) on aerial photography using Google Earth.

phone's internal sensing devices (such as its microphone) and external wireless sensors (such as off-the-shelf pollution sensors, health monitors, and GPS) to collect data. We focus on four of these prototypes here:

- PollutionSpy, a pollution monitor for urban areas;
- NoiseSpy, an on-device sound sensor;
- Fresh, a mobile forum for community environmental awareness; and
- MobAsthma, a mobile asthma and pollution monitor.

These prototype software components provide a robust and expandable platform for mobile sensing, including the technology for on-device data gathering, refining, integrating, and interpreting. They can also transfer data to and from remote servers linked to a broad range of resources. We hope our mobile system will have a direct impact on respiratory health and indirect impacts on healthy user habits, such as encouraging more walking and cycling.

Mobile Experiences

We ultimately want a system that lets users explore data and make informed decisions about how they interact with their environment, but it should also enable nontechnical users to use their mobile phones without specialist knowledge in large-scale sensor data collection in real time.

PollutionSpy

Our PollutionSpy application aims to monitor air pollution in traffic by using mobile phones to create a "pollution map" of Cambridge, England. It also promotes social networking in a local community through the provision of a Web portal that facilitates back-end sharing of real-time environmental and archived data.

PollutionSpy software creates a type of Bluetooth personal network and can connect up to seven different Bluetooth devices. Within this network, the mobile phone serves as the master, and the other devices function as slaves. So far, we've used this network to connect mobile phones to pollution sensors for CO, NO, NO_2, CO_2, and SO_2 as well as weather sensors for temperature and wind speed. Upon connection, the Bluetooth devices feed sensor data tagged with locations to a log file on the phone and display this collected data graphically on the phone's screen. Users also have the option to transfer the data to a remote database and view it in real time on our GIS mapping tools, which are embedded in a dedicated Web interface.

As Figure 1 shows, we've executed several data collection campaigns resulting in useful data. In these ex-

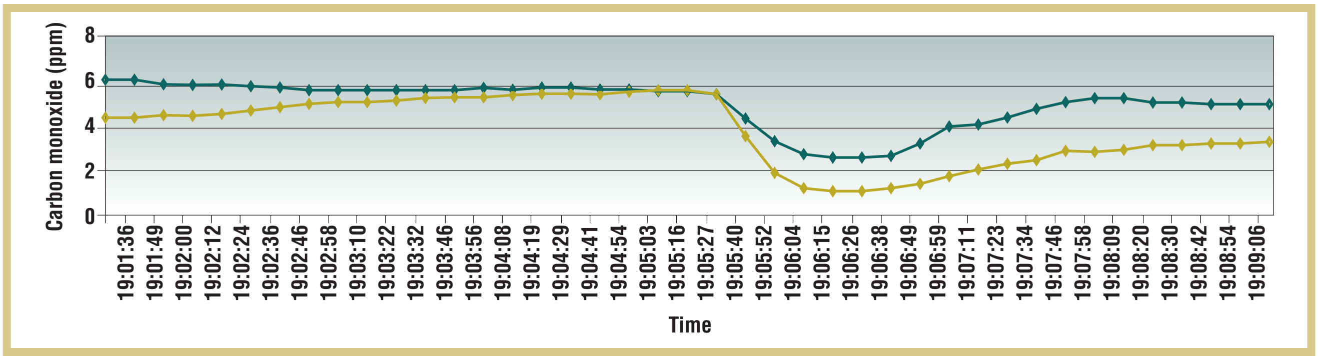


Figure 2. Comparison of two CO sensor readings. To test how closely matched the sensors are, we had two users walk down the same side of the road with sensor boxes.

periments, we focused on Nokia's third-generation (NSeries) N95 in a proof-of-concept trial to demonstrate that such monitoring techniques are reliable and provide useful data.[5]

The gas levels being measured can change very rapidly—some users reported that returning to the same geographical position a few minutes later yielded different results. To test how closely matched the sensors are, we performed several experiments with two users carrying sensor boxes down the same side of the road. As Figure 2 shows, the results indicate a very good level of coherence.

We recently developed more sensor boxes that we plan to calibrate and use in a large-scale experiment in Cambridge. In future trials, we hope to be able to answer questions such as how much variation a particular area has and over what scale do changes occur.

NoiseSpy

Sound is essential to our daily lives, but noise is not. Most people define noise as sounds that are loud, annoying, and harmful to the ear, and many people feel that traffic noise is one of the biggest offenders.

NoiseSpy is a sound-sensing system that turns mobile phones into low-cost data loggers for monitoring environmental noise. It lets users explore a city area while collaboratively visualizing noise levels in real time. The software combines sound-level data with external GPS receiver locations to generate a map of sound levels over the course of a journey. Each time the software runs on a mobile phone, it displays the noise data graphically on the phone screen along with other location information, as Figure 3 shows. We tested our initial NoiseSpy implementation with six cyclists over the course of two weeks. Initial feedback through participant diaries and interviews showed that the users generally enjoyed the experience. The software is currently available online (www.cl.cam.ac.uk/mobilesensing/downloads.htm).

As we expected, we observed that noise levels are higher in peak periods when roads are busy and lower in off-peak periods, and that noise levels over an area vary (for example, the Doppler effect, in which noise levels rise as a vehicle approaches and reduce again after it passes), which causes short-term variations in noise level. In our experiments, one participant walked past a quiet area and recorded a high noise reading because a car passed at that point. Also, high wind caused noise levels to rise near junctions and open areas.

Fresh

Fresh is a mobile interface that uses GSM networking and positioning via the cell IDs in users' phones to let people discuss "green" issues related to their local environment.[6] In addition, users can access environmental data points such as pollution levels, ambient noise, and weather events via the information other users post; the interface is a mobile phone tool, so it engages and encourages seamless participation in real time from multiple locations. Fresh could even help local communities improve their lifestyles by providing easy access to real-time, geographically measured environmental information.

In Fresh, the "world" is initially empty, but as the interactions start, the user's phone cell IDs fill up with questions and answers from other users making their way across the city. Users can search their current location for any information about the local environment or look at tagged questions and answers related to that area. They can then choose to answer any questions related to where they are with a short text response. If they don't find what they're looking for, they can start a new discussion by posting a question for others to answer. Figure 4 shows an example scenario of three users searching for, picking up, and answering environmental questions about their current location in Cambridge. Two of them want to know about ambient pollution levels.

The Fresh prototype is still under development. Our aim is to both motivate users and make the user interface more engaging and effective. Once the interface is mature enough, we plan to release the phone software and start experimenting with the system around Cambridge.

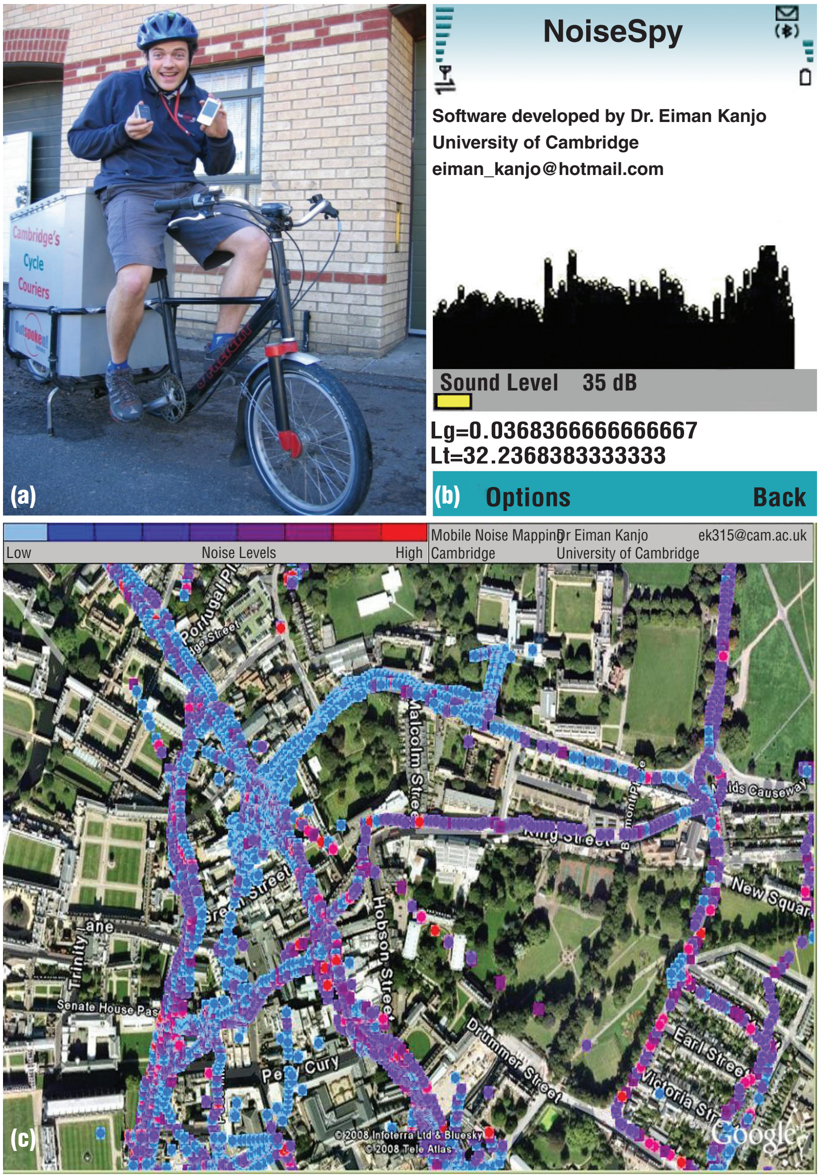


Figure 3. NoiseSpy. (a) A user carrying an N95 mobile phone with an external GPS ready to test the NoiseSpy application. (b) Screenshot of the NoiseSpy interface. (c) Google Earth visualization of noise data collected by participants from a local cycling courier company.

MobAsthma

As more of us live in urban areas, the ability to monitor and assess air quality becomes increasingly important for public health authorities. In particular, increased air pollution has been linked to increased rates of asthma.

MobAsthma is a personalized asthma-monitoring application prototype that lets asthma specialists and allergists investigate the relationships between personal exposure to air pollution and the prevalence of asthma and respiratory symptoms. Our smart phone application analyzes, in real time, measurements from medical devices such as asthma peak-flow and pollution data through wireless sensors and combines them with the patient's current location. The system is also capable of monitoring a person's asthma condition and remotely alerting medical staff if the patient experiences an asthma attack. This method could help people reduce their anxiety by managing their exposure to air pollution when levels are moderate or high. Figure 5 shows the MobAsthma application's system architecture and main components.

An initial trial of MobAsthma is under way in collaboration with Jon Ayres, a professor of environmental and occupational medicine at the University of Aberdeen.

Implementations

So far, we've implemented our phone software on N95 and N80 using native Symbian C++; we chose the latter because it lets phone software access all our development APIs.

MobSens software components installed on the phones must perform the following operations: sensing, filtering, processing, and logging sensor data; rendering screen displays, including graphs, maps, and user interfaces; and uploading data streams to backend servers in real time. Each data entry is combined with the last valid GPS location plus additional information such as latitude, longitude, speed, bearing, UTC date, time, the phone's International Mobile Equipment Identity (IMEI), user name, journey ID, and phone battery level.

Both PollutionSpy and NoiseSpy use a standard Bluetooth client-server architecture to receive data from external GPS units. We chose Bluetooth GPS because we tested our applications on both N95 and N80—N80 doesn't have a built-in GPS unit, and N95's built-in GPS has a fairly serious flaw. Specifically, the receiver is located under the number keypad, which means it can get a signal only when the phone is open. Even then, it takes a long time to get a signal, which isn't practical for our applications.

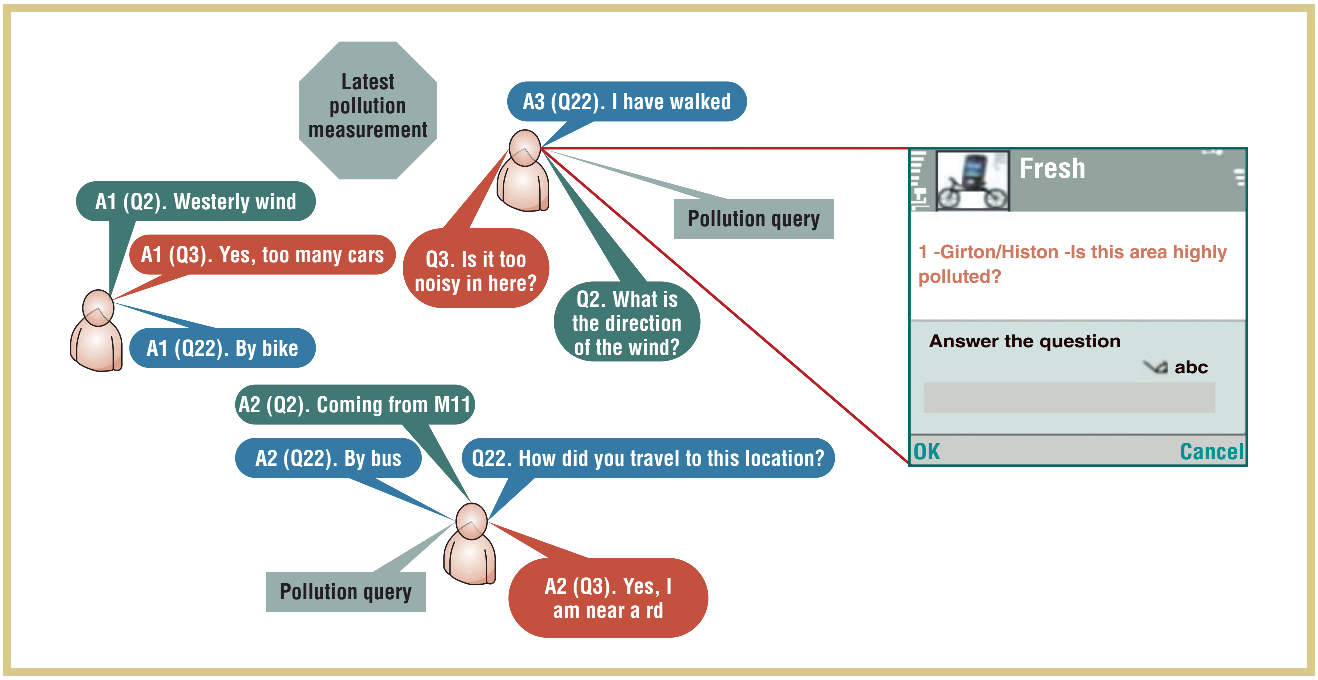


Figure 4. Fresh. Three people use the Fresh system in Cambridge to learn information about local environmental issues and events.

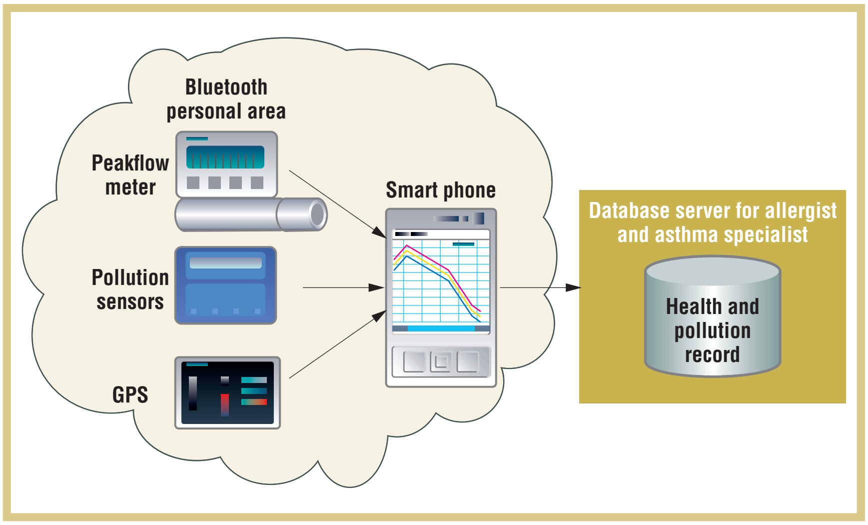


Figure 5. System architecture of MobAsthma application.

Currently, users have the following logging options for controlling how much data they want to store on the phone:

- interval (logging multiple times per day),
- intensive (logging very frequently, such as every second),
- regular (logging scheduled at regular intervals),
- real time (logging over GPRS only, on the phone only, or both GPRS and on the phone), and
- upload (logging files to the server once a day).

In Fresh, users' phone network cell IDs, questions, and answers are also attached to the data stream. Lookup tables of cell IDs include data for each cell tower's latitude and longitude, which are stored in the remote database.

Because most mobile phone networks don't provide mobile phones with routable IP addresses, all communication requests must be initiated from the client side. The client sends these calls to the server over HTTP using POST and GET requests, with the parameters passed within the POST request's data, then uses the reply to update the client application's state. The system uses POST to send information to the server, such as sensor data and user queries, and GET to obtain information from the server, such as local traffic information.

Lessons Learned and Research Challenges

Preliminary user studies for these various mobile application prototypes have revealed a range of research challenges that we must address.

Power Management

An important challenge when designing sensing applications on mobile phone platforms is power consumption—in particular, when the application uses multiple communication interfaces such as Bluetooth, GPS, Wi-Fi, and GPRS. We used Nokia's Energy Profiler for power management; this standard software tool specifically lets developers test and monitor their applications' energy usage in real time in the target device. We found that Bluetooth communications, screen display, sensor data processing, and GPRS radios were responsible for draining most of the battery power.

As application developers, we want to build tools that offer good fidelity and user experience without significantly altering a standard mobile phone's operational lifetime. One way to reduce mobile phone power consumption is to let users switch off screen displays and graphs at their own discretion. Similarly, we could add a timer to let users change the sampling rate from the sensors according to the task at hand, which might not require the frequent use of radios for communication or satellite signals for location coordinates. Finally, processing sensor data on mobile phones is computationally intensive and could consume a considerable amount of energy, potentially limiting the system's usability. With this in mind, we could do some basic data filtering on the phone and push the complex data-processing tasks to the back-end servers.

Processing sensor data on mobile phones is computationally intensive and could consume a considerable amount of energy, potentially limiting the system's usability.

Scalability

The MobSens framework has the potential to open a large-scale real-time environmental monitoring network around a city. However, such a development presents new and formidable concerns arising from the need to transmit, integrate, model, and interpret vast quantities of highly diverse spatially and temporally varying sensor data. For example, maintaining sensor reliability and software consistency across more than 70 MobSens nodes raises challenges in terms of dealing with failed mobile nodes and network links, potentially frequent updates from several users, and the impact of software updates on node reliability.

We're looking at ways to enable updating the phone application on users' handsets automatically, perhaps by adopting a client-server similar to MUPE (Multi-User Publishing Environment; www.mupe.net) in which users do a single install only and the client updates itself once new versions are available.

Handling Disconnections

Although connectivity is generally good in urban environments, we found that sensors still aren't connected 100 percent of the time. This intermittent connectivity leads to delay-tolerant sensing, in which data is cached for a time before it's uploaded when the connection is restored.

During our experiments, users experienced another form of wireless disconnection: the connection between the phone and sensors is subject to disconnect when the battery level is low on either the phone or the logger.

GPS signals suffer the same problems. Although there have been some advances in improving GPS receiver sensitivity and new techniques such as assisted GPS that permit a GPS receiver to use attenuated signals, a conventional receiver's antenna must have a direct line of sight to the GPS satellites. Our users reported some problems with GPS coverage in Cambridge due to the high urban canyons.

We're looking at ways to enhance our location-tracking techniques—for example, by working with HW-Communications (www.hwcomms.co.uk) to use its GSM-based cellular positioning alongside GPS tracking technology.

Privacy

Our users' top concern was the security and confidentiality of the data we collected about them. Almost everyone questioned the privacy and integrity of the sensed data itself, especially in correlation to the sensing devices. Some potential users refused to take part in some of our experiments because they feared their sensitive personal information would leak during data collection. However, many users trusted our system because they knew that we would guarantee their anonymity.

In MobSens, we chose to compromise the device's usability by designing the interface in a way that let the user modify the state of the communication and sensing modalities supported on the phone (audio, Bluetooth, and GPRS). Nevertheless, the system requires a better mechanism to let trusted users share sensing presence at high fidelity (and present a less accurate view to less trusted viewers).

Ongoing work in MobSens has begun to address these challenges by providing privacy policies that inform users about our data-handling practices up front and serve as the basis for a user's decision to release data. We're also hoping to develop efficient verification protocols that ensure secure data management and guarantee system integrity.

We're currently working on revising some of the components and improving a few architectural elements to reflect the valuable feedback from our study participants. Specifically, future revisions of the MobSens framework will include

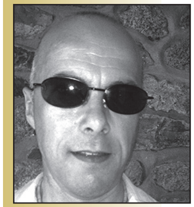
- an improved software module on the phone that prolongs battery life;
- an enhanced version of the Web portal, ideally one that reduces wait time and the amount of data shown to the user;
- an improved privacy policy setting as well as an enhanced user interface;
- a new way to handle mobile sensors and more advanced data aggregation algorithms;
- on-device visualization tools and mapping; and
- software that works with any smart phone on an open platform.

Visit the MobSens Project (www.cl.cam.ac.uk/mobilesensing) for the latest in mobile sensing research.

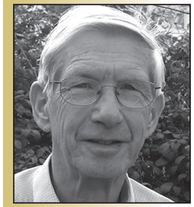
the AUTHORS

Eiman Kanjo is a researcher at the Computer Laboratory and Mathematical Sciences Centre at the University of Cambridge. Her main research interest is in mobile and pervasive sensing. Kanjo has a PhD from the University of Abertay Dundee in the area of pervasive and tangible interfaces based on computer vision and interactive tabletops. Her PhD work was patented in 2006. Kanjo is a member of the ACM. Contact her at eiman_kanjo@hotmail.com.

Jean Bacon is a professor of distributed systems at the Computer Laboratory, University of Cambridge, and a Fellow of Jesus College. She leads the Opera Research Group, with a focus on the design and deployment of open, large-scale, widely distributed systems. Bacon was the founding editor in chief of *IEEE Distributed Systems Online* and served as EIC of *IEEE Concurrency.* She is an IEEE and BCS Fellow and has served on the governing body of the IEEE Computer Society. Contact her at Jean.Bacon@cl.cam.ac.uk.



David Roberts is a research manager at Nokia. With his coding skills receding faster than his hairline, he recently jumped into technical management positions, the past seven years of which have been at Symbian. During his time at Symbian and Nokia, Roberts has headed up a consultancy group working closely with Nokia on S60 device creation before moving into research. Contact him at David.Roberts@nokia.com.



Peter Landshoff is professor emeritus of mathematical physics at the University of Cambridge, where he has been closely involved in the creation, and then the running, of several projects, including the Newton Institute, the Cambridge eScience Centre, the National Institute for Environmental eScience, the Cambridge Computational Biology Institute, the Centre for Quantum Computation, and the Millennium Mathematics Project. Contact him at pvl@cam.ac.uk.

ACKNOWLEDGMENTS

We thank our many collaborators on these experiences. We're also grateful to Mike Short and Ian Curran from O2 UK for providing unlimited GPRS airtime and Nokia for providing us with a large number of mobile phones for testing. We're also grateful to Symbian for its support.

REFERENCES

1. T. Weber, "How Smart Does Your Phone Need to Be?" bbcnews.co.uk, 8 Nov. 2007; http://news.bbc.co.uk/2/hi/business/7078877.stm.
2. J. Burke et al., "Participatory Sensing," *Proc. World Sensor Web Workshop*, ACM Press, 2006; www.sensorplanet.org/wsw2006/6_Burke_wsw06_ucla_final.pdf.
3. A. Campbell et al., "PeopleCentric Urban Sensing," *Proc. 2nd ACM/IEEE Ann. Int'l Wireless Internet Conf.* (WICON 06), ACM Press, 2006; http://portal.acm.org/citation.cfm?id=1234179.
4. E. Paulos, R. Honicky, and E. Goodman, "Sensing Atmosphere, Workshop on Sensing on Everyday Mobile Phones," *Proc. Conf. Embedded Networked Sensor Systems* (SenSys 07), ACM, 2007; http://repository.cmu.edu/hcii/203.
5. T. Simonite, "Cyclists' Cellphones Help Monitor Air Pollution," *New Scientist*, 2 Jan. 2008; http://technology.newscientist.com/article/dn13130-cyclists-cellphones-help-monitor-air-pollution.html.
6. E. Kanjo and P. Landshoff, "Fresh: CellID Based Mobile Forum for Community Environmental Awareness," *Proc. Ubiquitous Sustainability: Citizen Science & Activism*, Ubicomp, 2008; www.ubicomp.org/ubicomp2008/workshops.shtml.