ODK Sensors: an Application-level Sensor Framework for Android devices

Rohit Chaudhri, Waylon Brunette, Bruce Hemingway, Gaetano Borriello
Department of Computer Science and Engineering
University of Washington
Box 35350, Seattle WA, 98195, USA
{rohitc, wrb, bruceh, gaetano}@cse.uw.edu

ABSTRACT

To simplify interfacing a variety of external sensors with consumer Android devices, we developed a user-level framework that streamlines both application and driver development by providing abstractions that separate responsibilities between the user application, sensor framework, and device driver. These abstractions simplify the creation of sensing applications and provide for a high level of customization and flexibility, thereby enabling a variety of wired and wireless sensors to be connected to mobile devices. To avoid issues with modifying locked consumer devices, the framework's driver architecture is implemented at the user-level. To increase the variety of mobile sensing applications, we have developed a reconfigurable interfacing board that allows the framework to communicate with external sensors that have low-level digital or analog interfaces. We discuss field deployments that leverage the framework to address global health issues.

Keywords

Mobile computing, smart phones, ICTD, sensing, Bluetooth, USB, Open Data Kit

1. INTRODUCTION

Information technology applications are increasingly using mobile devices to build data collection systems. Sensors help bridge the physical and digital worlds by providing precise readings on various phenomena (e.g., location, oxygen level) that may be inaccurately described by humans. Integrating sensors directly within mobile applications can improve information flow as data can be automatically recorded avoiding errors associated with manual data entry. Additionally, difficulties such as training users to operate sensors and recognize bad data readings can be avoided. Unfortunately, building sensor-based Android applications can be challenging for developers because they need to handle threading/buffering, processing sensor-specific data, and the quirks of different communication channels. To lower the burden of connecting external sensors to Android devices we present Open Data Kit (ODK) Sensors, an application framework that simplifies the development of sensor-based mobile applications by decomposing monolithic mobile sensing applications into simpler, modular components.

DEV '13, January 11-12, Bangalore India

2. ARCHITECTURE SUMMARY

ODK Sensors hides complexities involved with interfacing external sensors to Android devices. The framework provides a common interface for developers to access both internal and external sensors (described in detail in [1, 4]). This abstraction creates a separation of concerns where top-level applications implement domain-specific use cases (e.g., an app to record patients' vital signs) that interface with the framework via a unified API that enables apps to discover sensors, connect/start/stop sensors, and receive sensed data. Additionally, the framework provides a simple abstraction with which to develop and deploy user-level device drivers. While a driver abstraction is a standard systems concept, ODK Sensors framework includes features that make development of drivers easier by automatically handling sensor state (e.g., channel management, connection, buffered data, threading) for driver developers, thereby reducing driver code to the handling of sensor-specific commands and data processing. This separation also enables app developers to focus on the application logic instead of sensor-specific logic.

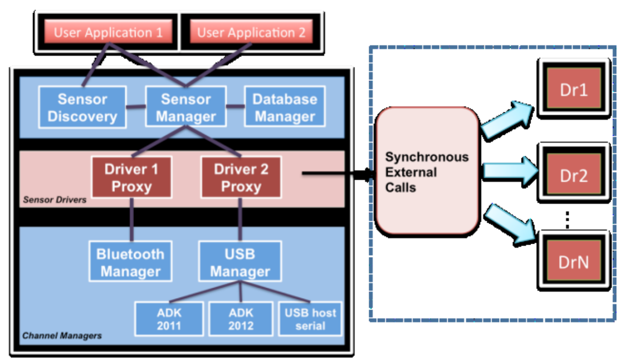

Figure 1: Architectural overview of ODK Sensors where a black rectangle represents a separate Android app. Top-level user applications leverage the framework's unified API to access sensor data. The framework communicates with sensor driver apps using Binder IPC (an Android service).

The framework can be used on any consumer Android device that may be 'locked' because the framework and drivers run at the application-level. To create a reusable ecosystem of sensor drivers, the framework is designed to leverage standard app marketplaces (such as Google Play) enabling users to find drivers and perform automatic updates the same as any Android app. Figure 1 shows the framework, sensor drivers, and top-level user applications are implemented as separate android apps. Sensing applications (e.g. User Application 2) communicate with the framework using synchronous calls exposed as an Android service. The framework communicates with sensor driver apps (e.g. DrN) via local driver proxies that provide sensor state management and encapsulate Android service call semantics over a synchronous service interface.

Isolating different components of a sensing system into separate apps allows them to be developed and maintained independently. This provides a high level of customizability and flexibility to end-users who can get a complete sensing system on their devices by aggregating the required components that may be written by different developers. After obtaining the sensing hardware, a user needs to download two android apps: 1) the ODK Sensors framework, and 2) a sensor driver. A top-level sensing application is then able to connect to the sensor and receive sensed data via the framework. To simplify development, the top-level application can leverage the framework's user interface to guide the user through sensor discovery and sensor driver selection.

In previous work [1, 4] the framework's USB channel only connected to an Arduino-based USB Bridge via the Android Accessory Development Kit (ADK 2011). However, recently more Android devices were released with increased USB capabilities including USB Host capabilities (e.g. Samsung Galaxy Nexus, Nexus 7). The ODK Sensors framework was expanded to support multiple modes of USB communication, including USB Host serial devices (e.g. FTDI), ADK 2011 complaint devices, and ADK 2012 complaint devices.

Motivating use cases and different types of mobile sensing applications for ODK Sensors were presented in [4]. To enable a wider variety of sensing applications we developed a custom interfacing board called FoneAstra (Figure 2) that connects to mobile devices over wired (UART) or wireless (Bluetooth) communication channels as well as sensors that only have low-level, analog or digital (e.g., I2C, SPI) communication interfaces. Being battery-powered, FoneAstra can operate autonomously enabling it to support applications like WaterTime [3], where a mobile device connects intermittently to a sensor board to retrieve sensed data buffered over a period of time. For applications like milk banking [5] and vaccine monitoring [2], in which a mobile phone is co-located with the board, FoneAstra has a 2nd micro-USB port that is used to power the co-located phone. When connected to a mobile device capable of being a USB Host the board can be powered by the mobile device itself and does not require a battery. Additionally, FoneAstra has an on-board SD memory socket, a real-time clock (for autonomous data collection), a few LEDs, and an audio buzzer for feedback to users.

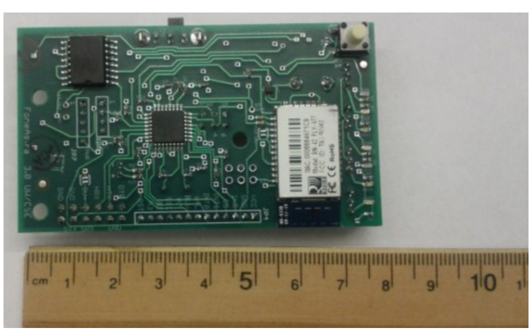

Figure 2: FoneAstra, a reconfigurable interfacing board to enable wired or wireless communications between mobile devices and low-level sensors connected to the board.

3. DEPLOYMENTS

Milk banks in South Africa are using ODK Sensors to monitor breast milk pasteurization. Milk banks pasteurize breast milk from donor mothers to deactivate pathogens (e.g., HIV, Hepatitis) before feeding vulnerable infants. We built a monitoring application (Figure 4) that connects a temperature sensor and a Bluetooth printer to an Android phone. The top-level user application utilizes the sensed milk temperature to guide users through the pasteurization process, and a successful completion prints a pasteurization report as well as labels for the milk bottles.

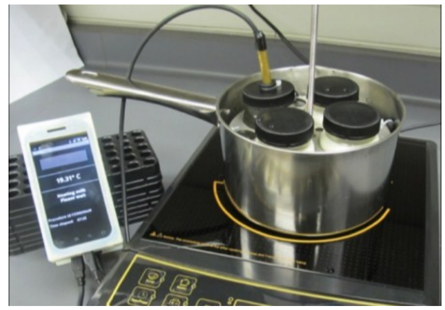

Figure 3: ODK Sensors deployment at a human milk bank in Durban. The application guides users through the pasteurization process by providing feedback based on temperature of breast milk (contained in jars that are heated).

In an upcoming clinical trial in India, ODK Sensors will be used to assist health workers diagnose childhood pneumonia through the integration of a pulse-ox sensor with a digital version of the Integrated Management of Neonatal and Childhood Illness protocol (IMNCI).

4. CONCLUSION

ODK Sensors is part of Open Data Kit's [6] modular set of tools that helps simplify sensing application development by creating a single interface to connect to both external and built-in sensors. The application-level driver framework enables convenient reuse of sensor-specific code between applications by logically separating the user application from the underlying sensor driver. All the components can be distributed through app markets.

5. REFERENCES

[1] Brunette, W., et al. "Open Data Kit Sensors: a sensor integration framework for android at the application-level". In Proc. of 10th Intl. Conf. on Mobile Systems, Applications, & Services, 2012.

[2] Chaudhri, R., et al. "Monitoring Vaccine Cold Chains in Developing Countries." IEEE Pervasive Computing, Volume 11, Issue 3. March 2012.

[3] Chaudhri, R., et al. "Sensors and Smartphones: Tracking Water Collection in Rural Ethiopia." IEEE Pervasive Computing, Volume 11, Issue 3. March 2012.

[4] Chaudhri, R., et al. "Open Data Kit Sensors: mobile data collection with wired and wireless sensors". In Proc. of the 2nd ACM Symposium on Computing for Development, March 2012.

[5] Chaudhri, R., et al. "A system for safe flash-heat pasteurization of human breast milk." In Proc. of the 5th ACM Workshop on Networked Systems for Developing Regions, 9-14, June 2011.

[6] Hartung, C., et al. "Open Data Kit: Building Information Services for Developing Regions". In Proc. of the 4th Intl. Conf. on Information and Communication Technologies and Development ICTD, Dec 2010.