Reflex: System and Programming Support for Efficient Sensor Use in Mobile Systems

Xiaozhu Lin[1] and Lin Zhong[1,2]
[1]Department of Computer Science and [2]Department of Electrical & Computer Engineering
Rice University, Houston, TX 77005

1. INTRODUCTION

Many emerging mobile services leverage sensors available on mobile systems to acquire information regarding the physical world. While the sensors themselves can be quite low-power, sensor use by mobile devices has proven to be a key bottleneck in the device energy efficiency due to a centralized and ad hoc system support. Our research in this area has led us to develop *Reflex*, a framework that manages sensor use with programming and operating abstractions called *channel*s for sensor data processing. Reflex provides three mechanisms. 1) It uses a runtime called channel manager to execute channels on behalf of applications and therefore eliminates frequent process context switches. 2) It allows multiple applications share a channel and therefore minimizes redundancy in sensor data processing. 3) It can delegate channel execution to programmable sensors and therefore relieves the system from high-duty cycle activities. Our measurement of Reflex prototypes based on N810 and N900 shows that Reflex can improve the system energy efficiency by 15%, 55%, and 90% with the three mechanisms, respectively.

2. DESIGN

In *Reflex*, channels are programming and operating abstractions for sensory information acquisition. A channel is developed as a code snippet that takes sensor data as input and generates sensory information as output, ranging from processed sensor data to recognized events. Reflex software consists of two major parts (see Figure 1): first, a *channel manager* on the host mobile system (e.g. smartphone) that accepts channel binaries from applications at run time, manages them, and dynamically delegates them to programmable sensors; second, a minimum *sensor runtime* on the programmable sensor that is responsible for sensor data acquisition, channel loading and execution, and communication with the host mobile system. More specifically, sensor runtime works with the support of a lightweight OS kernel. It executes each channel as a lightweight task and creates a special *proxy* task to 1) carry out administrative commands from the host system (e.g. loading channel, deleting channel, retrieving channel running status, etc.) and 2) report the acquired sensory information.

HotMobile '10, Feb. 22-23, 2010, Annapolis, M.D.

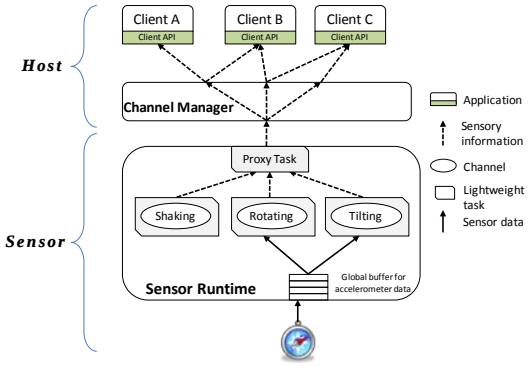

Figure 1. Sensory Information Acquisition in a decentralized Reflex System

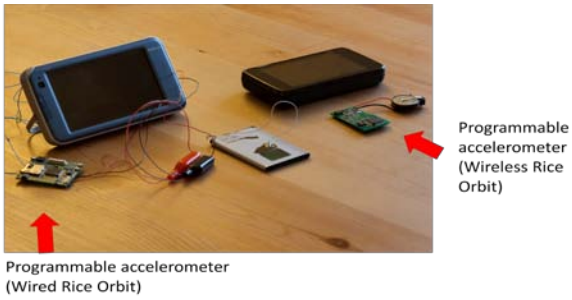

Figure 2. Reflex prototypes implemented with Nokia N810, N900, and two programmable sensors

3. PROTOTYPE

We have prototyped Reflex for Nokia N810 and N900, Linux-based mobile devices (Figure 2). We implement channel manager as a standalone system daemon that talks with applications based on d-bus IPC mechanism. Besides, we have implemented Reflex sensor runtime based on µC/OS-II, with both wired (RS232/UART interface) and wireless (Bluetooth) sensors, including Rice Orbit accelerometer sensor (with MSP430 as its microcontroller) and the CMU-Cam3 image sensor (with LPC2106 as its microcontroller).

In the demo, we will display our Reflex prototype with Rice Orbit sensors in action. Besides, we develop several demo channels, each of which detects a specific motion pattern, e.g. shaking and tilting. In particular, the Reflex-enhanced N900 can delegate one or more channels to Rice Orbit sensors and remain in deep sleep mode while the sensors stay alert of the physical world.

Project Homepage: http://Reflex.recg.rice.edu

Reflex is *GPL'd*: http://sourceforge.net/projects/RiceReflex/