A Prototype for Resource Optimized Context Determination in Pervasive Care Environments

Nirmalya Roy[1], Christine Julien[1], Archan Misra[2], and Sajal K. Das[3]

[1] Dept. of Electrical and Computer Engineering, The University of Texas at Austin
{nirmalya.roy,c.julien}@mail.utexas.edu
[2] Applied Research, Telcordia Technologies, New Jersey
archan@research.telcordia.com
[3] Dept. of Computer Science and Engineering, The University of Texas at Arlington
das@uta.edu

Abstract. In this demonstration we demonstrate an early prototype that shows the importance of context determination in the presence of mobile entities in pervasive care environments. We use our novel context model to build a framework for resource-constrained sensor networks. We then use this context model to use a user's mobility to infer his *activity*, which we refer to as his *context state*. Because the context state is inferred from actual sensed context, we use the prototype to demonstrate the tradeoff between context inferencing accuracy and communication overhead. The ability to sense the environment and accurately infer context can help monitor the user in a pervasive care environment.

1 Introduction

Because mobile applications operate in unpredictable and changing environments, the ability to sense and act on the changing context is essential to any application. An emerging scenario for mobile computing entails an assisted living environment, in which a mobile user and his surroundings are sensed by a variety of devices in the environment and carried by the user. These applications enable *aging in place*, promoting healthy independence. Energy-efficient determination of an individual's context (both physiological and activity) is an important technical challenge for assisted living environments. Given the expected availability of multiple sensors, context determination may be viewed as an estimation problem over multiple sensor data streams. We have developed a formal and practically applicable model to capture the tradeoff between the accuracy of context estimation and the overheads of sensing [2]. In particular, we use *tolerance ranges* to reduce an individual sensor's reporting frequency while ensuring acceptable accuracy of the derived context. In our vision, applications specify their minimally acceptable value for a Quality-of Inference (QoINF) metric. We introduce an optimization technique allowing our framework to compute both the best set of sensors to use to infer the particular context metric *and* their associated tolerance values, that satisfy the QoINF target at minimum communication cost. In this demonstration we show a prototype of this tolerance range based sensor data reporting system implemented on a SunSPOT sensor testbed.

S. Ystad et al. (Eds.): MobiCASE 2009, LNICST 35, pp. 335–338, 2010.

2 Implementation

We use the SunSPOT [1] (Sun Small Programmable Object Technology), a small, wireless, battery-powered experimental platform. We employ two types of devices: *free-range* devices that are embedded in the environment and perform sensing and communication tasks, and a *base-station* device that simply communicates with the free-range devices and is connected to a larger format device (e.g., laptop) for supporting a user interface. Each free-range SunSPOT contains a processor, radio, sensor board, and battery, and the base-station SunSPOT contains the processor and radio only. The SunSPOT uses a 32-bit ARM9 microprocessor running the Squawk VM and programmed in Java, supporting IEEE 802.15.4 radio. In our demonstration we use the accelerometer sensor available with SunSPOT sensor board.

Our demonstration includes two scenarios. In Case 1, the free-range devices reports each data sample to the base-station regardless of the user's actual requirements. This provides the highest quality information but also incurs the highest cost. Using this sensor data, our application infers the user's context state, such as sitting, walking, or running. In Case 2, we use an application-specified QoINF to limit the sensors' data reporting frequencies to reduce the communication overhead while still meeting the target accuracy for estimating the context state. In this case, the sensor reports only if the change in the sensed value is outside of the sensor's specified tolerance range. This is a simple example with intuitive behavior and expected performance outcomes. The goal of the demonstration is to exercise our QoINF programming framework and demonstrate how application developers can have a significant impact on the lifetime of their network by trading accuracy for overhead.

The sensor application on the free-range device contains two java class files, one to maintain the radio communication connection and another to read and send the accelerometer data. The host application on the base-station consists of three main Java classes, one to handle the radio connection to the remote SPOT sensor, one to display the data collected, and the third to manage the GUI.

2.1 Case 1: No Tolerance Range

In this first case, we do not ask the application to specify any tolerance, and the motion sensor constantly reports the data samples used to infer the user's context state. We use the measured acceleration to determine the context state. If the person is sitting, the acceleration due to the earth's gravity will be $1g$ along the positive Z-axis, and $0g$ along the X and Y axes. By monitoring the deviation of acceleration from the $1g$ of gravity, we can infer whether the user is in motion. As seen in Figure 1, we can easily confirm the different context states from this measurement (sitting ($1g$), walking ($2.5g$), or running ($4.5g$)). We have shown how this information can be used to support a variety of context-aware tasks in an assisted living environment [2].

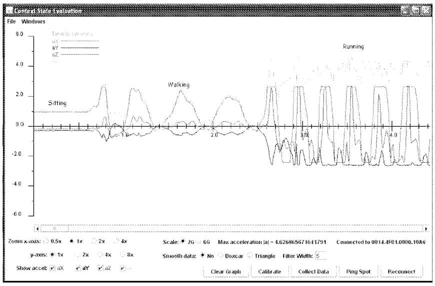

Fig. 1. Context Sensing with Accelerometer

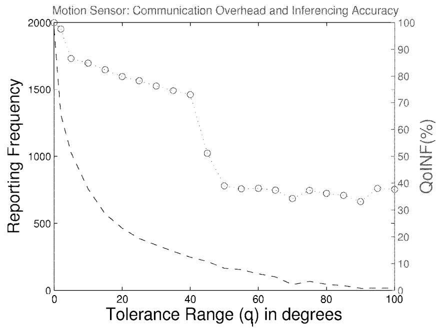

Fig. 2. Overhead and Accuracy vs. Tolerance

2.2 Case 2: Using Tolerance Range

In the second case we consider a tolerance range derived from a QoINF function provided by the application, which causes the sensor to report only if changes in the sensed values exceed the given tolerance range. This reduces the communication between the free-range sensors and the base-station, ultimately increasing the life of the energy constrained devices. We use the following simple logic to reduce the sensor's communication overhead.

```
SendSensorData (input range) {
  last_reported_valuex=acc.getRawX();
  last_reported_valuey=acc.getRawY();
  last_reported_valuez=acc.getRawZ();
  //value of x, y and z-axis acceleration fluctuation in voltage
  if (ABS(last_reported_valuex-acc.getRawX()) >= range OR
      ABS(last_reported_valuey-acc.getRawY()) >= range OR
      ABS(last_reported_valuez-acc.getRawZ()) >= range){
       Datagram.writeByte(acc.getRawX());
       Datagram.writeByte(acc.getRawY());
       Datagram.writeByte(acc.getRawZ());
       //sending the packets or reporting the data samples
  }
}
```

2.3 Measuring QoINF Accuracy and Communication Overheads

To study the potential impact of varying the tolerance ranges, we collected traces for the SunSPOT motion sensor for a single user engaged in a mix of three activities (*sitting*, *walking*, and *running*) for a total of $\approx$ 6 minutes (2000 samples at $5.5Hz$). We then used an emulator to mimic the samples that a sensor would have reported for different tolerance ranges and compared the context inferred in both cases to the ground truth. Figure 2 shows the total number of samples reported (an indicator of the reporting overhead) and the corresponding QoINF achieved (defined as $1 - \textit{error rate}$) [2] for different values of the tolerance range.

A QoINF accuracy of $\approx 80\%$ is achieved for a modestly large tolerance as shown in Figure 2. Moreover, using this tolerance range reduces the reporting overhead dramatically. This suggests that it is indeed possible to *achieve significant savings in bandwidth, if one is willing to tolerate marginal degradation in the accuracy of the sensed context.*

3 Demonstration Details

In the demonstration we will show two SunSPOTs reporting data samples for the user in the same state for both of our cases. In Case 1 we show the remote sensor reporting every data sample regardless of any tolerance range or user context state. This data gathering process is demonstrated by a continuous graph at the base-station as shown in Figure 1 and with a steady blue LED at the free-range sensor.

In Case 2 the sensor reports its data sample only if there is a tangible change. The data gathering process is demonstrated by a graph at the base-station and with a flashing blue LED at the free-range sensor. We do this for three different SunSPOTs sensor each using a different tolerance range (0, 20, and 100). For tolerance range 0, the blue light will flash constantly, similar to Case 1. In this case we achieve a near 100% inferred context state accuracy. For tolerance range 20, the blue light flashes only with a change in the context state and for 100 it flashes rarely, even with a frequent change in the context state. This confirms that, with a medium tolerance range, we can achieve a moderate context accuracy with much reduction in overall sensor sample reporting frequency, whereas a high tolerance range can nullify the reporting overhead but is very error prone.

Acknowledgments

This work was funded in part by Air Force Office of Sponsored Research (AFOSR), Grants #FA9550-09-1-0155 and #FA-9550-07-1-0157. The conclusions herein are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

References

1. Project SunSPOT. Sun Small Programmable Object Technology, `http://www.sunspotworld.com`
2. Roy, N., Misra, A., Das, S.K., Julien, C.: Quality-of- Inference (QoINF)-Aware Context Determination in Assisted Living Environments. In: Proc. of WiMD 2009 in conjunction with MobiHoc 2009 (May 2009)