AN ARCHITECTURE FOR MOBILE PHONE SENSING

Tadej Vodopivec, Boštjan Kaluža, Simon Kozina, Mitja Luštrek
Department of Intelligent Systems
Jožef Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
e-mail: tadej.vodopivec@ijs.si

ABSTRACT

This paper presents a sensing system that logs sensor data on an Android-based mobile device and stores it to a remote server for further processing and analysis. This system consists of an application for Android operating system, FTP file server, a PostgreSQL database, an application that reads data from FTP and populates the database, an application that processes data stored on the server, and a web application that displays captured data.

The main purpose of this system is to provide sensing capabilities for applications that provide supervision or assistance with activities of daily living, coordination of services by outside health care providers, or monitoring of user activities to help to ensure their health, safety, and well-being.

KEYWORDS

Sensors, logging, monitoring, ambient assisted living, mobile phone sensing.

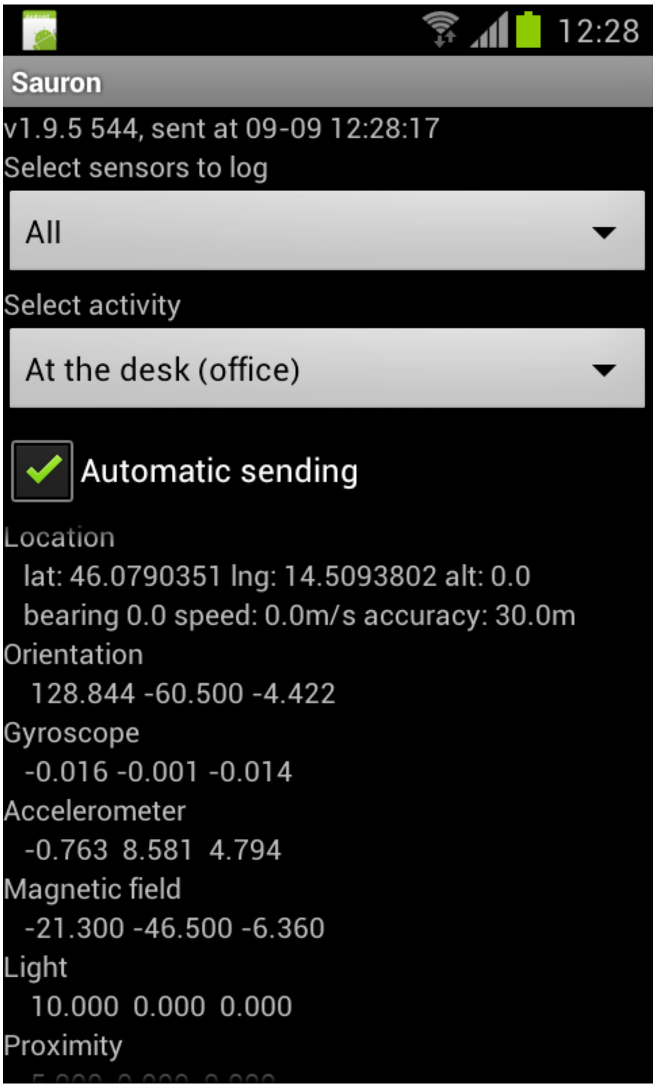
1 INTRODUCTION

By 2050, one in every three persons living in the more developed regions is likely to be 60 or older, and about one in every four is projected to be 65 or older [1]. Many of them will need help and there will be even less young people to help them than today. Yet most of them will have electronic devices with them, and we can use those devices to help them.

The aim of human monitoring system is to track user activity, detect interesting events and optionally trigger actions or alarms when needed. This kind of system is helpful in healthcare, ambient assisted living, baby care and many other areas.

This paper describes the Sauron system, an application that collects sensor data on an Android-based mobile device, and stores the data to a database for real-time processing or off-line analysis. The application will (1) speed up the process of creating and testing applications that rely on real world mobile sensing, and (2) provide training data for modelling user activities.

A few examples where the Sauron can be applied: an application that monitors person's everyday life and makes a suggestion if necessary, for example, if you take a walk every morning through the park and one day there is a construction site in the way, the application would alert you before you even left the house and suggest a detour.



v1.9.5 544, sent at 09-09 12:28:17
Select sensors to log
All
Select activity
At the desk (office)
☑ Automatic sending
Location
lat: 46.0790351 lng: 14.5093802 alt: 0.0
bearing 0.0 speed: 0.0m/s accuracy: 30.0m
Orientation
128.844 -60.500 -4.422
Gyroscope
-0.016 -0.001 -0.014
Accelerometer
-0.763 8.581 4.794
Magnetic field
-21.300 -46.500 -6.360
Light
10.000 0.000 0.000
Proximity

Figure 1: Main view of Android application user interface with status bar at the top and displayed data in the bottom.

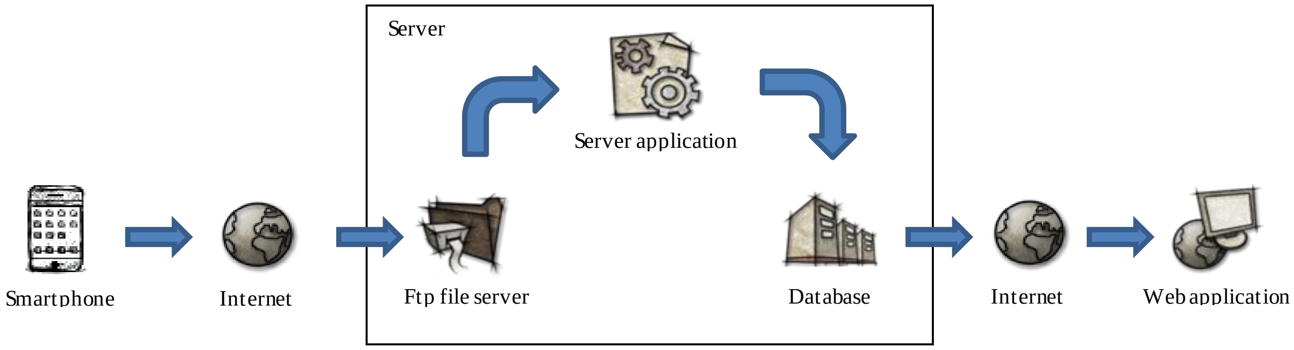


Figure 2: System architecture and its components [7]

Another example would be for specific tasks like fall detection [2], to recognize user's activity and estimate daily energy expenditure [3], or for detection of unusual behaviour based on analysis of daily-living dynamics [4].

The paper is structured as follows. Section 2 will introduce the system architecture and its components. Next, Section 3 described the sensor data – available at a mobile device. Finally, Section 4 concludes the paper.

2 SYSTEM STRUCTURE

System consists of five separate components as shown in figure 2. Each component is an application of its self:

- Android application – gathers sensor data on a mobile device,
- FTP server – used to temporary store the data before it is written to the database,
- Server application – reads the data from FTP server and populates the database,
- Database – where all data is stored and labelled,
- Web application – visualizes data from database.

Part of the applications run on the server side with the following requirements:

- Windows XP or newer
- PostgreSQL server 8.3 or newer
- Java 6 or newer

Android application runs as a client with the following requirements:

- Smartphone with Android 2.3.3 or newer
- GPS, gyroscope and accelerometer sensors
- WiFi internet access recommended

2.1 Android application

Sensor data is acquired with an Android application from a smartphone running Android operating system (shown in Figure 1). It can log and store data from all sensors available to operating system.

Most interesting data is supplied by acceleration sensor, orientation sensor, gyroscope sensor and magnetic field sensor. Application can also log location based on two different services. If GPS signal is available, it logs GPS coordinates. If GPS location is not available (for example inside buildings), it logs coordinates based on a buildings electromagnetic footprint using Qubulus' Cloud Positioning API for Android [5], which estimates the position within building. This is done by creating a radio frequency fingerprint of a building and comparing it to measurements on the device.

The application is designed to be used both in a laboratory environment and in real world. On startup, it requires basic information about user and type of activity being recorded. This is required, to label the data, which is important in longer and complex activities. Then, it starts a service that records all selected sensors and stores the sensor data to a SD card on the smartphone. If it is connected to internet via Wi-Fi, it periodically sends all logged data to FTP server. Wi-Fi is recommended because the application generates big amount of data, but 3G or any other internet connection is supported.

2.2 FTP server

Due to security reasons Sauron cannot communicate directly with an external database, hence the data is stored temporary on a FTP server, and afterwards into a database by an application running on the server side.

2.3 Server application

Now that all data is located on FTP server, system has to permanently store it to database. This is done by an application that reads raw text files, connects to database and stores the data in tables corresponding to particular sensor type. This way it is ensured that only our verified application could modify the database.

2.4 Database

For data storage we used PostgreSQL server. There we stored information for every time a sensor value has

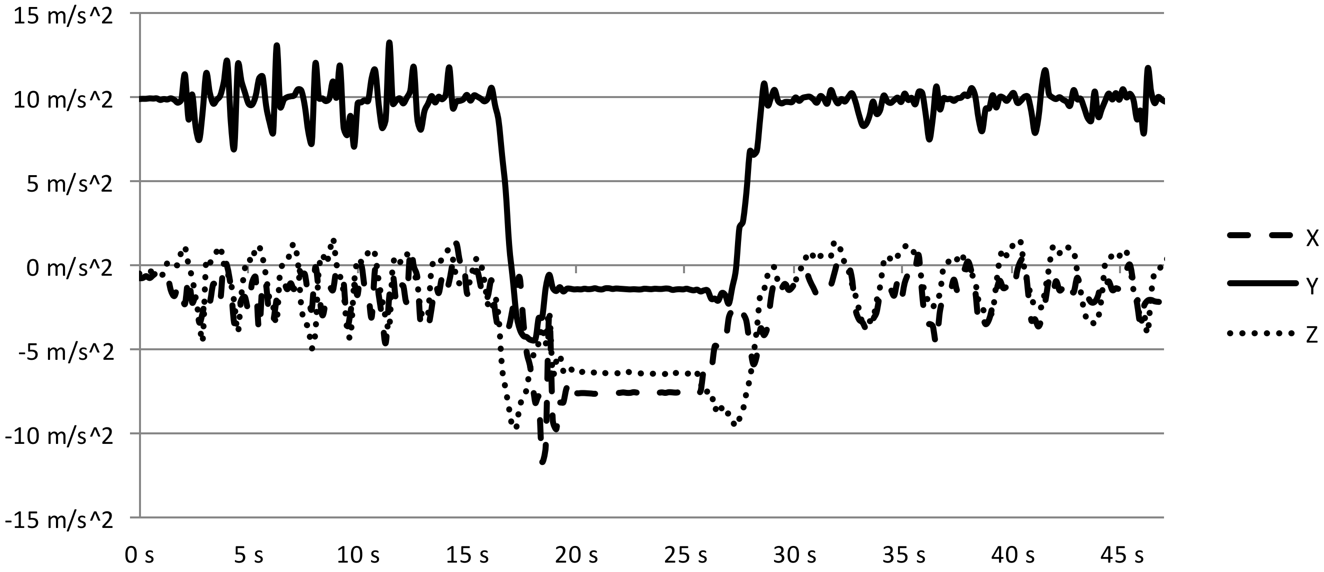


Figure 3: Accelerometer data over time. In this example, subject was walking, then lying down and then at around 30 s started walking again. Gravity acceleration is clearly visible as Y value (solid line).

changed. This includes: timestamp, user, scenario activity and sensors value at the time.

2.5 Web application

At this point all data is stored in the database and we can use any external application to process it. In this example, we developed a web application that plots the data on a chart as seen in Figure 3 or display it on a map.

3 ANDROID SENSORS

Most of the Android devices have a number of sensors attached to the device. Some sensors provide raw values, some values are calculated, some have a coordinate system relative to the device, and some have coordinate system relative to planet Earth. This section will describe how to understand values returned by Android sensors.

3.1 Coordinate system

Accelerometer, magnetic field, gyroscope, gravity and linear acceleration sensors use a coordinate system relative to the telephone. The axes always stay the same, even if the orientation of the telephone changes. Axes point in directions as shown in Figure 4.

Rotation vector and orientation sensor use a coordinate system relative to the Earth as shown in Figure 5.

3.2 Raw sensor data

The data are provided directly from the sensor. Values are stored in an array, where first number is value in direction X, second number is value in direction Y and third number is value in direction Z.

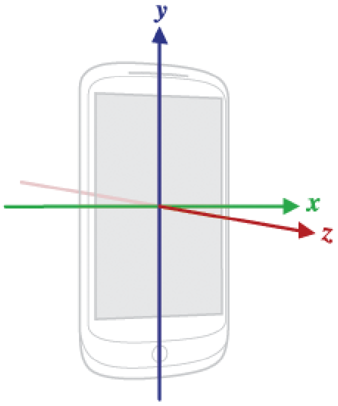


Figure 1: Axes in phone coordinate system [6]

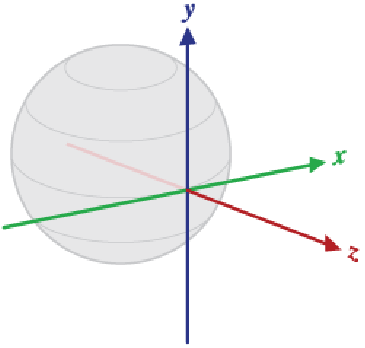


Figure 2: Axes in Earth coordinate system [6]

The unit of measurements are as follows: values for accelerometer are in m/s^2; values for magnetic field are in µT (microtesla); values for gyroscope are in /s (radian per second), where rotation is positive in the counter-clockwise direction, and the output of the gyroscope could be integrated over time to calculate a rotation describing the change of angles over time; value for light is in SI lux (lm/m^2); value for proximity sensor is in centimetres (cm). Some proximity sensors only support a binary near or far measurement. In this case, the sensor should report its maximum range value in the far state and a lesser value in the near state.

3.3 Calculated sensor data

Some values are calculated from values of other sensors to give some new or better information. An example is gravity sensor that shows the direction of gravity acceleration. If the device is stationary, values should be same as accelerometer.

Linear acceleration indicates acceleration along each axis, not including gravity.

The rotation vector represents the orientation of the device as a combination of an angle and an axis, in which the device has rotated through an angle θ around an axis <x, y, z>. The three elements of the rotation vector are <x*sin(θ/2), y*sin(θ/2), z*sin(θ/2)>, such that the magnitude of the rotation vector is equal to sin(θ/2), and the direction of the rotation vector is equal to the direction of the axis of rotation.

4 CONCLUSION

System described in this paper logs values from sensors on an Android smartphone and saves them in a database. This paper focuses on data acquiring and storage, and omits higher-level computation.

The system performs well in logging sensor data, but it generates huge amounts of network traffic and has high demand on server hardware, if it is used for real time processing by multiple clients.

We successfully used it for laboratory measurements with ten users for recognizing basic activities and we combined measurements with other sensors. This will allow us to create a smartphone application that will inform the user about the energy expenditure. There are also on-going measurements for more complex activities for detecting unusual behaviour that benefits from inside and outside location.

Such a system has great potential for use in healthcare or any other area where real-time mobile sensing is required.

5 References

[1] World Population Ageing: 1950-2050 , http://www.un.org/esa/population/publications/worldageing19502050/ (Accessed September 21, 2012)

[2] Gjoreski, H., Luštrek, M., and Gams, M., Accelerometer placement for posture recognition and fall detection. Intelligent Environments conference, 2011.

[3] Luštrek, M., Cvetkovic, B., and Kozina, S, Energy expenditure estimation with wearable accelerometers. ISCAS conference, 2012.

[4] B. Kaluža and M. Gams, Analysis of Daily-Living Dynamics. Journal of Ambient Intelligence and Smart Environments, 2012.

[5] Qubulus | Indoor Positioning Service Provider, http://www.qubulus.com/ (Accessed September 8, 2012)

[6] Android Developers, http://developer.android.com/reference/packages.html (Accessed September 7, 2012)

[7] Kreski -Lines- by ~azazel100, http://azazel100.deviantart.com/