Share-Lock Power Saver Energy Efficiency for Smartphone Using Sensors

M. Bhadkumbhe[1], Manisha M.Jagtap[2], Rupali G.Ghadge[3],Devashree A.Fegade[4], Pratiksha S.Ghodekar[5]

[1] *Professor, Computer Engineering, PDEACOEM, Maharashtra, India*
[2] *Student, Computer Engineering, PDEACOEM, Maharashtra, India*
[3] *Student, Computer Engineering, PDEACOEM, Maharashtra, India*
[4] *Student, Computer Engineering, PDEACOEM, Maharashtra, India*
[5] *Student, Computer Engineering, PDEACOEM, Maharashtra, India*

ABSTRACT

Nowadays smart phone are used context-awareness which is increasingly important for a range of mobile and pervasive application. Human-centric con- text have been extensively researched those are used in indoor/outdoor, at home/in office, driving/walking. Phone perspective is used on table/sofa or in pocket/bag/hand. Micro-Environment sensing platform is design and implement that automatically records sensor hints and characterizes the micro- environment of smart phone. Through programming interfaces smart phone provides fine-grained environment information to upper layer application.It is a unified framework covering the major cases of phone usage, placement, attitude, and interaction in practical uses with complicated user habits. As a long-term running middleware, it considers both energy consumption and user friendship. The preliminary results show that it achieves low energy cost, rapid system deployment, and competitive sensing accuracy. We are going to make use of that data by reading and converting it into ASCII format. We will be developing various applications using that data for security as well as for saving the battery of mobile. The sensors which we are going to use in our project are Accelerometer, Light, Pressure, Proximity etc.

Keywords:- *Mobile Phone sensing, activity Recognition, power management*

1. INTRODUCTION

In Micro-Environment Sensing for Smartphones, we use different mobile sensors like Accelometer, camera, Touchscreen, Pressure Sensor etc. To use the data broadcasted by the sensors in order to make useful applications in security and optimization domain. In this project we are going to develop application which supports the security and energy consumption. In this project using the mobile sensors like proximity, accelerometer, touchscreen, camera etc, we are going to create the different modules. Using these sensors we developing the module which support the security and increase our mobile battery life. The aim of micro-environment sensing on smartphones is to provide a more general primitive for novel human centric applications, especially in healthcare and behavior monitoring. Identifying the phones micro-environment also opens new possibilities to perform fine grained context-aware energy saving strategies, which is essential for battery powered smartphones. On detecting being placed in the drawer, for instance, it is reasonable for the phone to infer that it will not be used in the near future, and can switch to certain power saving mode and turn off unnecessary sensors and software.

2. LITERATURE SURVEY

[1]Title: Battery monitoring and analysis for Android based system.

Author: Swapnili P. Karmore, Anjali R. Mahajan, Suruchi Kitey.

In this module, we would find out the place where mobile is placed. We would check the condition of mobile, whether it is in hand or kept on some surface. We would be doing this by using Environment, Metal Detector, Magnetic Field Detector sensor. If it is found that mobile is not in use, then by stopping the running processes we could save thebattery. Once the mobile is back to active mode we will start those processes.

[2]Title: Privacy control in smart phones using semantically rich reasoning and context modeling.

Author: Dibyajyoti Ghosh, Anupam Joshi, Tim Finin and Pramod Jagtap

we will identify the soft surfaces by using metal detector sensor. If call arrives on soft surface and mobile is on vibration mode then in that case application will switch on the ringer mode so that user will come to know about the incoming call as soft surfaces does not give vibration sense.

[3]Title: Energy Management Techniques in Modern Mobile Handsets.

Author: Narseo Vallina-Rodriguez and Jon Crowcroft, Fellow

we will be generating Morse code using flash sensor. We have to type a word that we need to generate in an application and flash sensor will do the rest of the task.

[4]Title: Energy Efficient and Continuous Environment Sensing Android Applications

Author: Deepak Todkar1, Sajal Wani2, Adarsh Vaidya3, Sadhna Sisode4

We need to check the code on different android version to make it compatible for all the devices. We need to write an adaptive parser algorithm which will change for each sensor as per the data size transfer of sensor.

3. PROPOSED SYSTEM ARCHITECTURE

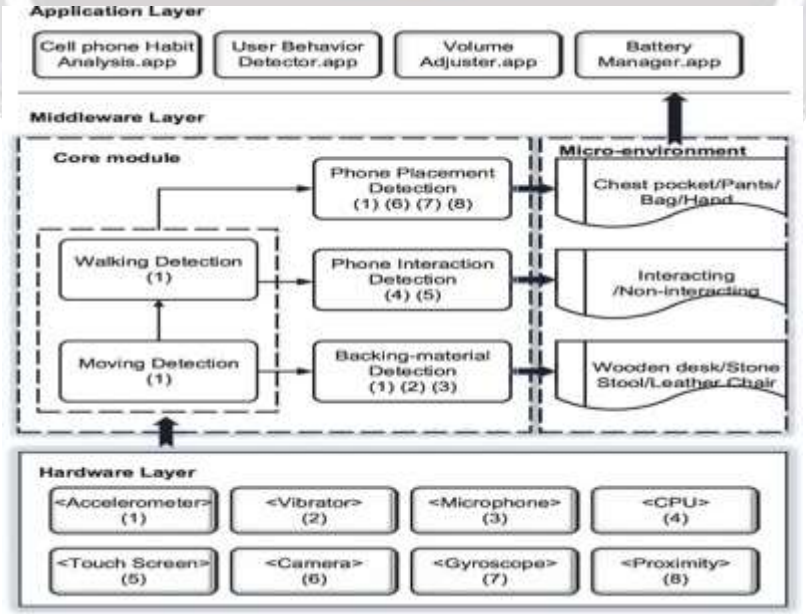

fig:-System Architecture

In this application we are going to use proximity sensor. We would be checking open and close conditions of proximity sensor. Suppose mobile is placed in the pocket or in closed environment, then proximity sensor will be close. Application should not receive call at that time. We will check Close Open Close condition at that time .If mobile is placed in an Open environment then we will pick up the call for Open, Close condition of proximity sensor. Pressure Sensor .In this application, we are using touch and pressure sensor of screen to measure the pressure on a single point of screen. If that pressure is greater than the threshold pressure of the application then it will trigger the alert to the configured numbers in an application. Wrong screen unlock location tracker: If someone enters a wrong pattern lock then a picture would be taken of him/her using the front camera and then we would be latching his location using GPS or LBS. We will send this location, time and image taken to the configured Email ID. If front camera is absent we will only send location and time to configured Email ID

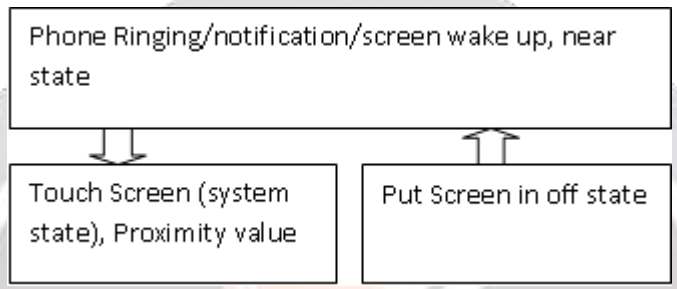

Fig:-Flow Diagram

4. ALGORITHMS USED

4.1 Message Digestion (MD5)

The MD5 algorithm is a widely used hash function producing a 128-bit hash value. Although MD5 was initially designed to be used as a cryptographic hash function, it has been found to suffer from extensive vulnerabilities. It can still be used as a checksum to verify data integrity, but only against unintentional corruption. Like most hash functions, MD5 is neither encryption nor encoding. It can be cracked by brute-force attack and suffers from extensive vulnerabilities as detailed in the security section below. MD5 was designed by Ronald Rivest in 1991 to replace an earlier hash function MD4.[3] The source code in RFC 1321 contains a "by attribution" RSA license. The abbreviation "MD" stands for "Message Digest." The security of the MD5 has been severely compromised, with its weaknesses having been exploited in the field, most infamously by the Flame malware in 2012.

4.2 Message-Digest algorithms characteristics

Message-Digest (Fingerprint) algorithms are special functions which transform input of (usually) arbitrary length into output (so called "fingerprint" or "message digest") of constant length. These transfsormation functions must fulfil these requirements:

1. No one should be able to produce two different inputs for which the transformation function returns the same output

2. No one should be able to produce input for given prespecified output Message-Digest algorithms serve in digital signature applications for guaranteeing consistency (integrity) of data. Commonly used model is as follows (message-digest in cooperation with asymmetric cryptography):

1. Sender creates input message (M) and computes its message digest (sMD). Then he uses his private key and encrypts message digest (esMD).

2. Encrypted message digest (esMD) is attached to the input message (M) and the whole message (M-esMD) is sent to receiver.

3. Receiver gets the message (M-esMD) and extracts the encrypted message digest (esMD). Then he computes his own message digest (rMD) of the received message (M). He also decodes received message digest (esMD) with sender's public key and gets decoded message digest (desMD). Then he compares both message digests (rMD ?= desMD). When both message digests are equal, the message was not modified during the data transmission.

4.3 MD5 algorithm description

MD5 algorithm takes input message of arbitrary length and generates 128-bit long output hash. MD5 hash algorithm consist of 5 steps:

Step 1. Append Padding Bits

Step 2. Append Length

Step 3. Initialize MD Buffer

Step 4. Process Message in 16-Word Blocks

Step 5. Output

5. FUTURE SCOPE

1]In this project we are going to focus on 10 different modules in which we are developing auto call picker, mobile security as well as user security, Environment changer, battery optimizer. User Classes and Characteristics: The users of this application are normal people having android smart phone.

2]Operating Environment: Android device we are using must have all the sensors needed for the application to perform well. Android 2.2 and above is needed.

3]Design and Implementation Constraints: Need a mobile handset with all the sensors present to perform the tasks.

4]User Documentation: User will be provided with all the user manuals and system information documents.

5]Assumptions and Dependencies: The user is expected to use android mobile with OS 2.2 or above. The mobile should have proximity, accelerometer, magnetic field detection sensors to broadcast the data.

6. CONCLUSION

We are going to present the design for micro-environment sensing for smartphones via collaboration among built-in sensors.Using mobile sensors we are going to develop the application for security and battery saving. We club various sensors in this application of those result achieves low energy and competitive Micro-Environment Sensing Accuracy.The platform automatically collects sensor hints and characteristics the immediate surroundings of smartphones providing environment information to application.

7. REFERENCES

[1]. 1.J. Yang, S. Sdhom, G. Chandrasekaran, T. Vu, H. Liu, N. Cecan, Y. Chen, M. Gruteser, R. Martin, "Detecting Driver Phone Use Leveraging Car Speakers", *Proc. ACM MOBICOM*, 2011.

[2]. S. Nath, "ACE: Exploiting Correlation for Energy-Efficient and Continuous Context Sensing", *Proc. 10th Int'l Conf. Mobile Systems Applications and Services (MobiSys '12)*, 2012.

[3] T. Yan, D. Chu, D. Ganesan, A. Kansal, J. Liu, "Fast App Launching for Mobile Devices Using Predictive User Context", *Proc. 10th Int'l Conf. Mobile Systems Applications and Services (MobiSys '12)*, 2012.

[4].C. Qin, X. Bao, R.R. Choudhury, S. Nelakuditi, "Tagsense: A Smartphone-Based Approach to Automatic Image Tagging", *Proc. Ninth Int'l Conf. Mobile Systems Applications and Services (MobiSys '11)*, 2011.

[5]. H. Lu, W. Pan, N.D. Lane, T. Choudhury, A.T. Campbell, "Soundsense: Scalable Sound Sensing for People-Centric Applications on Mobile Phones", *Proc. Seventh Int'l Conf. Mobile Systems Applications and Services (MobiSys '09)*, 2009.

[6]. H. Lu, J. Yang, Z. Liu, N.D. Lane, T. Choudhury, A.T. Campbell, "The Jigsaw Continuous Sensing Engine for Mobile Phone Applications", *Proc. Eighth ACM Conf. Embedded Networked Sensor Systems (SenSys '10)*, 2010.

[7].M. Azizyan, I. Constandache, R. Choudhury, "SurroundSense: Mobile Phone Localization via Ambience Fingerprinting", *Proc. ACM MOBICOM*, 2009.

[8].A. Rai, K. Chintalapudi, V. Padmanabhan, R. Sen, "Zee: Zero-Effort Crowdsourcing for Indoor Localization", *Proc. ACM MOBICOM*, 2012.

[9]. P. Zhou, Y. Zheng, Z. Li, M. Li, G. Shen, "IODetector: A Generic Service for Indoor Outdoor Detection", *Proc. 10th ACM Conf. Embedded Networked Sensor Systems (SenSys '12)*, 2012.

[10]. X. Zhu, Q. Li, G. Chen, "APT: Accurate Outdoor Pedestrian Tracking with Smartphones", *Proc. ACM INFOCOM*, 2013.