International Journal of Engineering and Advanced Technology (IJEAT)
ISSN: 2249-8958 (Online), Volume-6 Issue-4, April 2017

Development of Sensors on Android Platform

Akhilesh Kumar Sharma, Brijendra Kumar Sharma

Abstract- Mobile phones play increasingly bigger role in our everyday lives. Today, most smart phones comprise a wide variety of sensors which can sense the physical environment. In this research, we propose and demonstrate my DAM4GSN architecture to capture sensor data using sensors built into the mobile phones. Specifically, we combine an open source sensor data stream processing engine called 'Global sensor n/w (GSN)' with the android platform to capture sensor data. We present the design, implementation, evaluation, and user experiences of the Cence-me application, which represents the first system that combines the inference of the presence of individual using off-the self sensor enabled mobile phones with sharing of this information through social networking applications such as face-book and my-space. An android based application that monitors the vehicle through an On Board Diagnostics (OBD-2) interface, being able to detect accidents.

Keywords: DAM4GSN architecture, Cence-Me application, On Board Diagonstics(OBD-2) interface.

I. INTRODUCTION

Mobile phones have built-in sensors that can be used to measure different parameters such as motion, position and various environmental conditions. The sensing capability and data processing power of mobile phones have been increased during the past decade. My main contribution, DAM4GSN architecture, allows GSN to capture context annotated sensor data from low level computational devices such as mobile phones without porting GSN itself into mobile phones. A widespread adoption of mobile telephony has remarkably improved interpersonal communications in our society. Nowadays mobile phones resemble small multifunction computers, being characterized by a CPU power and RAM size similar to that of laptop computers available only a few years ago. The future trend is that more and more users own these intelligent mobile terminals, and that their main use gradually shifts towards functionalities including web surfing, social networking, multimedia streaming, online games, demotic applications, and so on. Under these premises, it is possible to introduce novel services using smart-phones in many different contexts. The emerging demographic change towards an ageing population is introducing drastic into our society. Nursing homes and care facility units are renowned solution for elderly people. A person who lives in these units becomes depress due to lack of independence. In both developed and developing countries of smart-phone users are increasing day by day.

Manuscript published on 30 April 2017.
***** Correspondence Author (s)
Akhilesh Kumar Sharma, M.Tech. Scholar, Department of Robotics and Automation, Gurugobind Singh Indraprastha University, Dwarla Sector 16, New Delhi-110078, India.
Brijendra Kumar Sharma, M.Tech. Scholar, Department of Robotics and Automation, Gurugobind Singh Indraprastha University, Dwarla Sector 16, New Delhi-110078, India.

For example, there are more than 16 million smart phones owned by Koreans and 100% of the population has access to a mobile phone network. Smart-phone runs a complete operating system and provides a platform for application developers and users. Google Android is one of the most competitive markets due to its open source platform. Hundreds of application have been developed ranging from the interactive games to healthcare domain. Especially the medical domain applications enable the users to interact with the system to provide real time user assistance and help to improve the people life's style.

II. CENCE ME IMPLEMENTATION

In this section, we present the CenceMe implementation details. The CenceMe application and system support consists of a software suite running on Nokia N95 mobile phones and backend infrastructure hosted on server machines. The software installed on the phones performs the following operations: sensing, classification of the raw sensed data to produce primitives, presentation of people's presence directly on the phone, and the upload of the primitives to the backend servers. Primitives are the result of: i) the classification of sound samples from the phone's microphone using a discrete Fourier transform (DFT) technique and a machine learning algorithm to classify the nature of the sound; ii) the classification of on board accelerometer data to determine the activity, (e.g., sitting, standing, walking, running); iii) scanned Bluetooth MAC addresses in the phone's vicinity; iv) GPS readings; and finally, v) random photos, where a picture is taken randomly when a phone keypad key is pressed or a call is received. Classification algorithms that infer more complex forms of sensing presence (i.e., facts) run on backend machines

Phone Software- Figure 1 shows the CenceMe architecture for the Nokia N95 phone. The phone architecture comprises following software component.

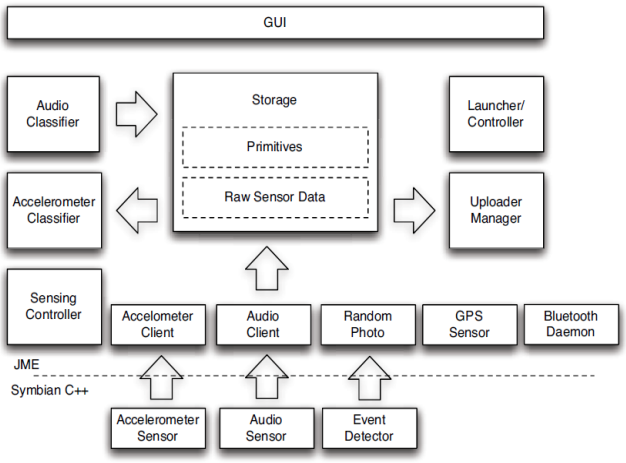

Figure-1

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Symbian Servers- The accelerometer sensor, audio sensor, and event detector sensor are Symbian C++ modules that act as daemons producing data for corresponding JME client methods. Their function is, respectively: polling the on board accelerometer sensor, sampling the phone's microphone, and detecting incoming/outgoing calls and keypad key presses. The sensed data is sent to the JME methods through the socket. Event detected by the event detector daemon are used by the random photo module at the JME level to generate random pictures, to trigger a photo upon an incoming phone call or to signal the application that is has to restart after a phone call for reliability reasons.

Bluetooth Daemon- This component resides at the JME level and is used to perform an enquiry over the Bluetooth radio to retrieve the MAC addresses of any neighboring Bluetooth nodes. The MAC addresses of neighboring nodes are used to determine if there are Cence Me phones in the area at the time of enquiry.

Accelerometer Client- This component is written in JME and connects through a socket to the accelerometer sensor to retrieve the accelerometer data byte stream. The byte stream is stored in local storage and retrieved by the activity classifier to compute the activity classifier to compute the audio primitive.

Audio Client- This JME client component connects through a socket to the Symbian audio server to retrieve the audio byte stream that carries the PCM encoded representation of the sound sample. The byte stream is stored in local storage and retrieved by the audio classifier to compute the audio primitive.

Random Photo- This JME module is designed to trigger the capture of the photo upon detection of incoming calls or pressed keypad key. The events are received through a socket from the event detector daemon. When the picture is taken it is stored locally until the next upload session.

GPS- The JME GPS implementation supplies a callback method that is periodically called by the Nokia GPS daemon to provide the geographical location of the phone. The GPS coordinates are stored locally and then uploaded to the backend servers.

Sensing Controller- This component is responsible for orchestrating the underlying JME sensing components. The sensing controller starts, stops, and monitors the sensor clients and the Bluetooth manager and GPS daemon to guarantee the proper operation of the system.

Local Storage- This component stores the raw sensed data records to be processed by the phone classifiers. As the classification of raw data records is performed, the data records are discarded; hence none of the sampled data persists on the phone. This is particularly important to address the integrity of the data and of the privacy of the person carrying the phone since none of the raw sensed data is ever transferred to the backend. Primitives, GPS coordinates, and Bluetooth scanned MAC addresses are stored in local storage as well, waiting for an upload session to start.

Upload Manager- This component is responsible for establishing connections to the backend servers in an opportunistic way, depending on radio link availability which can be either cellular or WiFi. It also uploads the primitives from local storage and tears down the connection after the data is transferred. Details about how the upload manager interacts with the backend.

Privacy Settings GUI- The privacy settings GUI allows the user to enable and disable the five sensing modalities supported on the phone, (viz. audio, accelerometer, Bluetooth, random photo and GPS). Users can control the privacy policy settings from the phone and the CenceMe portal. By doing so users determine what parts of their presence with or not as the case may be.

Click Status- To complement the full visualization of current and historical sensing presence available via the CenceMe portal, we developed ClickStatus, visualization client that runs on the mobile phone. The sensing presence is rendered as both icons and text on the phone GUI. The presence rendered by ClickStatus is subject to the same privacy policies settings as when viewed using the CenceMe portal. After a user logs in with their CenceMe credentials, they are presented with a list of their CenceMe buddies downloaded from the CenceMe server. CenceMe buddies are Facebook friends running Cence Me on their N95. While this is always done at start up, a user has the ability to refresh their buddy list at any time via a menu command option. By highlighting and selecting a buddy from buddy list, a user triggers ClickStatus to fetch via GPRS or WiFi the latest known sensing presence for the selected buddy from the CenceMe server. This presence is displayed on a separate result screen; from there a user can either exit to return to their buddy list or refresh the currently displayed buddy's presence.

Watch Tasks- The purpose of Watch Tasks is to restart any process that fails. Watch Tasks also serves several other ancillary purposes including: i) launching CenceMe when the phone is turned on; ii) starting the CenceMe application software components in the correct order; iii) restarting the CenceMe midlet after a phone call is complete. This is detected when the event detector daemon exits, signaling the end of a call; iv) restarting all support daemons when CenceMe fails. Such action is necessary when we cannot reconnect to specific daemons under certain failure conditions; and finally v) restarting all the CenceMe software components at a preset interval to clear any malfunctioning threads. The CenceMe phone suite uses a threaded architecture where each JME component shown in Figure 1 is designed to be a single thread. This ensures that component failure does not compromise or block other components.

Backand Software- The CenceMe backend software architecture is shown in Figure 2. All software components are written in Java and use Apache 2.2 and Tomcat 5.5 to service primitives from phones and the application requests from the CenceMe portal, ClickStatus, and Facebook. Communications between the phone and the backend uses remote procedure calls implemented by the Apache XML-RPC library on the server. Requests are handled by Java servlets in combination with a MySQL database for storage.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

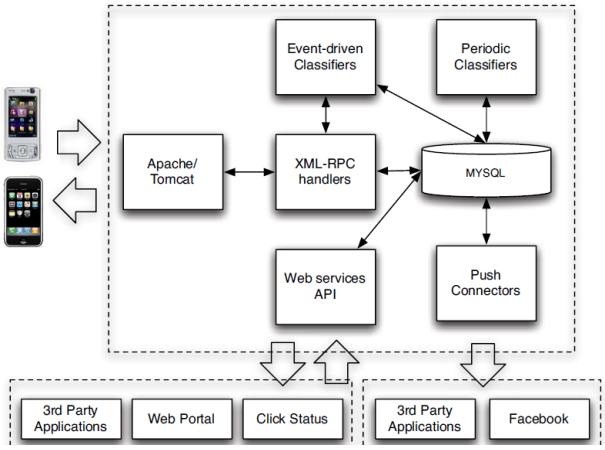

Figure 2- Software Architecture of the CenceMe Backend

III. MOBILE PHONE LIMITATIONS

OS Limitations- Although top-end mobile phones have good computational capability, often including multiple processors, they are limited in terms of the programmability and resource usage control offered to the developer. For example, the Nokia N95 is equipped with a 330 MHz ARM processor, 220 MHz DSP, and 128 MB RAM. However, when developing a non-trivial application on mobile phones a number of challenges arise. This is due in part because mobile phones are primarily designed for handling phone calls in a robust and resilient manner. As a result, third party applications running on the phone may be denied resource requests and must be designed to allow interruption at any time so as not to disrupt regular operation of the phone. This places a heavy burden on application exception handling and recovery software. While programmers may expect exception handlers to be called rarely, in Symbian they are called often and are critical to keeping an application and the phone operational. At the same time, testing exception handlers is difficult because a voice call can interrupt application code at any point in its execution; OS induced exceptions are outside the control of the programmer.

API and Operational Limitations- Additional limitations arise from the APIs provided by the phone manufacturers. JME implements a reduced set of the Java Standard Edition APIs for use on mobile phones. Because each phone model is different even from the same manufacturer, the Symbian OS and JME must be ported to each phone which typically results in missing or malfunctioning APIs for important new or existing components, such as an accelerometer or GPS. These API limitations may not be resolved by the manufacture because new models replace old models in quick succession. As a result, the programmer is forced to come up with creative solutions to API limitations. Examples of such API limitations and operational problems encountered with the N95 include a missing JME API to access the N95 internal accelerometer and JME audio API that exhibits a memory leak, respectively.

Security Limitations- To preserve the phone's integrity and protect the cellular network from malicious attacks, phone manufacturers and cellular network operators control access to critical components, including the APIs for access to the file systems, multimedia features, Bluetooth, GPS, and communications via GPRS or WiFi, through a rights management system. Properly signed keys from a Certificate Authority are needed to remove all restrictions on using these APIs.

IV. MOBILE PHONE SENSORS

In this paper we focus on the Android platform and Android enabled mobile phones. Most modern mobile phones have variety of different sensors built into them. These sensors can be divided into three main categories: Motion sensors (accelerometer, gravity, gyroscope, linear accelerometer, and rotation vector), Position sensors (orientation, geomagnetic field, and proximity), and Environment sensors (light, pressure, humidity and temperature). Even though, the Android platform supports twelve different sensors, the mobile phones may not have all these sensors built into them. Sensors built into a mobile phone could be varied depending on the hardware manufacturer. There are few other sensors built into the mobile phones: microphone as the audio sensors, camera as the video sensor, digital compass, and Global Positioning System (GPS) sensors. In this paper, we do not consider these types of sensors. The sensing capability of a mobile phone is limited to few sensors, it can be easily extended by using Personal Area Network (PAN) technologies such as IrDA, Bluetooth, Wireless USB, Z-Wave, Near field communication (NFC) and ZigBee. Pan technologies can connect additional sensing devices to the mobile phones.

V. GLOBAL SENSOR NETWORK

The Global Sensor Network is a platform aimed at providing flexible middleware to address the challenges of sensor data integration and distributed query processing. It is a generic data stream processing engine. GSN has gone beyond the traditional sensor network research efforts such as routing, data aggregation, and energy optimization. The design of GSN is based on four basic principles: simplicity, adaptivity, scalability, and light weight implementation. GSN middleware simplifies the procedure of connecting heterogeneous sensor provides to applications. Specifically, GSN provides the capability to integrate, discover, combine, query, and filter sensor data through a declarative XML-based language and enables zero-programming deployment and management. The above reasons lead us to choose GSN as our data processing engine over other alternative solutions.

VI. DAM4GSN ARCHITECTURE

We discuss the proposed Data Acquisition Model For GSN (DAM4GSN) architecture by dividing it into three separate sections: server configuration, client configuration, and data formats. The server configuration section explains how the GSN server needs to be configured in order to collect sensor data from mobile phones. The client configuration section explains how the mobile phone needs to be configured in order to read the sensor data through built-in sensors and send them to the GSN server.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

The data format section explains how the communication between GSN server and mobile phone can be done and how the data packets are formatted.

VII. UBIQUITOUS HEALTH MONITORING

Wearable health monitoring systems integrated into a ubiquitous mobile health system (mHealth), emerged as a technology of choice for ambulatory monitoring. This approach facilitates continuous monitoring as a part of a diagnostic procedure, optimal maintenance of a chronic condition, or computer assisted rehabilitation. Traditionally, personal medical monitoring systems, such as Holter monitors, have been used only to collect data for off-line processing. Ubiquitous connectivity allows real-time processing and communication: it also facilitates warnings, computer assisted rehabilitation, and continuous health monitoring.

VIII. ACTIVITY TRACKING ARCHITECTURE

We investigate the requirements in terms of communication, storage, processing and smart phone development platform to make it an acceptable solution. It is divided into three layers smart home, cloud computing and application layer.

Smart Home Layer- Smart home is ubiquitous sensing technology to recognize and track the daily life activities of inhabitants. Sensors are deployed on different objects and locations to sense the environment. It periodically sends the collected data to the server. For its processing, many machine learning and probabilistic models are developed for recognize the daily life activities.

Smart home is an intelligent agent that perceives its environment through different kind of sensors. We deployed RFID Tags, Biosensors, Micaz and Masol on different objects and locations to track inhabitant's daily routines. Our developed smart home has certain overall goals, such as minimizing the cost of maintaining the home and maximizing the comfort of its inhabitants. To achieve these goals, we take the advantage of new style of computing paradigm.

Cloud Computing Layer- Cloud computing is new paradigm that can provide dynamically scalable and virtualized resources as a service with pay-as-you-go manner. By pooling the various life care IT resources into clouds, hospitals can reduce the cost and increase utilization as the resources are delivered only, when they are required. We utilized cloud computing as Infrastructure as a Service (Iaas) to provide processing power and storage space for sensory data. A smart home sensed the environment and sends the data to cloud that is stored in a database as a sensor log. On cloud, our activity recognition algorithms process the sensors logs intelligently to recognize the daily life activities and evaluation of these algorithms are available.

Application Layer- We have developed Elderly Person Reminder Application and Care Giver Assistant Application for tracking of the daily life activities. The details are given bellow.

1-Elderly Person Reminder Application: Elderly people lives alone in their own homes and perform daily life activities. There are possibilities to involve in more than one activity and also a chance of forgetting the previous one. Our smart-phone based application is attention capturing application and activated when a critical running activity needs to be complete first for avoiding any sort of accident. For example: a person started cooking activity and engages in watching television, so running stove may cause an accident. In this case, developed application notifies the elderly person about incomplete activity with attention gaining music. It also generates soft reminders to do certain activity at specific time. If the attention to complete critical activity is ignored more than three times by the elderly person, then the application notify care giver or family member for assistance.

2-Care Giver Assistant Application: The smart-phone application also facilitates the family members of elderly person to keep track their activities when they are outside from home. The goal of this application is to enhance the level of awareness of care givers by notifying then when critical activity left unintended. The notification activate customized application interface with elderly person picture and current atmosphere readings. In order to make sure, the elder take medicine dosage according to the schedule to ensure medicine works effectively and illness is properly treated.

IX. VEHICULAR MONITORING AND SAFETY APPLICATION

The proposed application basically combine two elements vehicle and smart-phone- to achieve a symbiosis between both that is able to improve the effectiveness of emergency services by making accident detection fully automatic. Accident detection is based on the parameters provided by the OBD-II interface, such as airbag triggering detection, and is complemented with the information gathered by the mobile phone itself, such as GPS information. Initially the smart-phone connects to an OBD-II device via Bluetooth to retrieve data from the vehicle's bus. The information gained; together with data from the other sources (e.g. GPS system) is packed and sent to an emergency services database or to other third parties defined by the user if an accident is detected. This procedure is followed by an automatic call to an operator, which will send an ambulance or other rescue services to the accident location.

X. DIFFERENT WAYS OF DEVELOPING MOBILE BUSINESS APPLICATION

Before beginning the development of a mobile business application, one has to consider the basic development type for the application. It determines the scope of provided features available in the application. In addition to this, based on the chosen type of application development, aspects like programming language (e.g. Java or HTML5 and JavaScript) or architectural design patterns (e.g. Model-view-Controller) will be basically determined (Heitkotter et al. 2013). In the following, we define four different development types for mobile applications which have to be distinguished.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)

Web Applications: This type of mobile business application, e.g. jQuery Mobile Applications, is implemented using HTML5, CSS, and JavaScript. Usually, the application code is rendered within a mobile web browser. According to this, these applications are independent of a platform and device. Compared to native applications, this is a lightweight approach. In contrast to their minimal implementation effort, such applications provide the smallest set of functionally, because they rely on the feature set of web browser.

Hybrid Application- Web: The application code of this type mobile business application, e.g. Adobe Phone-Gap Applications (Adobe Systems Inc. 2013), consists of the same web technologies as for real Web Applications. However, they run in a native application container (e.g. iOS Web View), which, in turn, can provided additional functions.

Hybrid Applications- Mixed: These type of Hybrid Applications e.g. Appcelerator Titanium Applications (Appcelerator Inc. 2013), provide a core of mobile development APIs which will be normalized across platforms. These APIs can be used like known web technologies (cf. Hybrid Applications- Web). However, during the deployment of the HTML5 and elements of the corresponding native platform must be automatically performed.

Native Applications: Native Applications, e.g. Android Applications, represent the way of implementing mobile business applications using the standard API. They are implemented using the native programming language (e.g. Java for Android), and therefore require more special knowledge of device-specific development issues. They, in turn, offer the most possible functionality through direct use of respective platform provided programming libraries and interfaces. In addition, native applications show the best performance and applicability for users.

REFERENCES

1. K. Aberer, M. Hauswirth, and A. Salehi. Infrastructure for data processing in large-scale interconnected sensor networks. In Mobile Data Management, 2007 International Conference on, pages 198–205.
2. L. Cai, S. Machiraju, and H. Chen. Defending against sensor-sniffing attacks on mobile phones. In Proceedings of the 1st ACM workshop on Networking, systems, and applications for mobile handhelds, MobiHeld '09, pages 31–36, New York, NY, USA, 2009. ACM.
3. Crossbow Technology Inc. Crossbow-manuals getting started guide. Technical report, Crossbow Technology, September 2005.
4. F. Fitzek, M. Pedersen, G. P. Perrucci, and T. Larsen. Energy and link measurements for mobile phones using ieee802.11b/g. In Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops, 2008. WiOPT 2008. 6th International Symposium on, page 36, april 2008.
5. Google Inc. Android developer guide: Sensors, 2011. http://developer.android.com/guide/topics/sensors/index.html [Accessed on: 2011-12-26].
6. GSN Team. Global sensors networks. Technical report, Ecole Polytechnique Federale de Lausanne (EPFL), 2009.
7. GSN Team. Global sensor networks project, 2011. http://sourceforge.net/apps/trac/gsn/ [Accessed on: 2011-12-16].
8. P. Guillemin and P. Friess. Internet of things strategic research roadmap. echnical report, The Cluster of European Research Projects, 2009.
9. P. Klasnja, S. Consolvo, T. Choudhury, R. Beckwith, and J. Hightower. Exploring privacy concerns about personal sensing. In Proceedings of the 7th International Conference on Pervasive Computing, Pervasive '09, pages 176–183, Berlin, Heidelberg, 2009. Springer-Verlag.
10. G. Kortuem, F. Kawsar, D. Fitton, and V. Sundramoorthy. Smart objects as building blocks for the internet of things. Internet Computing, IEEE, 14(1):44–51, 2010.
11. N. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. Campbell. A survey of mobile phone sensing. Communications Magazine, IEEE, 48(9):140 –150, sept. 2010.
12. M. Lennighan. Total telecom: Number of phones exceeds population of world, May 2011. http://www.totaltele.com/view.aspx?ID=464922 [Accessed on: 2011-12-30].
13. Salehi. Design and implementation of an efficient data stream processing system. PhD thesis, Ecole Polytechnique Federale de Lausanne (EPFL), 2010.
14. H. Sundmaeker, P. Guillemin, P. Friess, and S. Woelffl´e. Vision and challenges for realising the internet of things. Technical report, European Commission Information Society and Media, 2010.

Published By:
Blue Eyes Intelligence Engineering and Sciences Publication (BEIESP)