LOOC: A CYBER-PHYSICAL SOCIAL NETWORK ON ANDROID PLATFORMS

By

CARMEN PATRICIA AYERDIS ESPINOZA

Bachelor of Computer Systems

Universidad Católica

Managua, Nicaragua

2000

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December, 2010

LOOC: A CYBER-PHYSICAL SOCIAL NETWORK ON ANDROID PLATFORMS

Thesis Approved:

Dr. Xiaolin Li

Thesis Adviser

Dr. Subhash Kak

Dr. Blayne Mayfield

Dr. Mark E. Payton

Dean of the Graduate College

ACKNOWLEDGMENTS

I would love to express my gratitude to all the people who support me during this amazing two years. I am very grateful with Fulbright Fellowship for this life experience. Also, my thanks go to Dr. Li for the opportunity to work with him and be part of his amazing team of students. It was such an exciting experience to be part of the computer science family at Oklahoma State University. I have never met such a faculty so excited to teach their students how to be better and more dedicated.

Special thanks to the wonderful family God gave me. My gratitude goes to my baby brother Jose Luis for being such a wonderful source of inspiration. Thanks to God for giving me the best American sister I could possible ask, Stacey Bridges who welcomed in her house and her life. Thanks to Noah Paul Evans for making me smile during this year. While studying in OSU I met amazing people and friends….Thanks to everybody who supports me.

Finally, to my dad Dr. Guillermo Ayerdis thanks, you were right…… this country is an amazing one and I still miss you. This work is dedicated especially to you

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Figure……Page

CHAPTER I

INTRODUCTION

According to the International Telecommunication Union (ITU), the increase of new mobile subscribers has grown approximately 24 percent each year between 2000 and 2008. It has been projected that by the end of 2008, 61 percent of the total population worldwide will be using mobile phones, an estimated four billion people[1]. In the United States, in the 1990's 34 million people used this technology to communicate. Currently, more than 203 million people are cell phone users in the US [2].

Thanks to the growing use of mobile devices, the applications for mobile phones are emerging as one of the areas that are developing fast in popularity and difficulty. Also, the complexity and demand for new features are increasing. Due to these factors, the Open Handset Alliance, a collection of 47 companies was created in 2007, and together developed Android, an open-source operating system for mobile devices to enhance the accessibility of mobile devices software creation. Some important members of this group are Google, T-Mobile, Motorola, E-bay and others[3].

At the same time, social networks are growing fast. Facebook, just one of the most popular social networks website claims it has over 200,000,000 users. Social networks can be defined as a set of groups of people that are interconnected to each other according to similar interest [4]. This powerful tool working in conjunction with mobile technology can be used to create interesting applications that millions can use. This is an interesting option for marketing and the creation of new products targeting people with the desired to stay connected and informed constantly about their connections information update.

One goal for this study is to do research in both areas combining both powers to design and evaluate Android applications for cyber-physical social networks.

1.1 Purpose of the study

The purpose of this study seeks to research and describe the use of Android mobile applications for physical social networks. Also, this study explores topics related to Android mobile device and online social networks by developing an application using the Android Software Development Kit (SDK). In addition, this paper will help to develop understanding of issues in Android applications and online social networking with emphasis in the physical environment.

1.2 Objectives of the study

This proposal has the goal of doing research in issues related to both Android mobile applications and online social networks. The following are some of the general objectives for this paper:

- Develop a cyber-physical application in Android for social network.
- Provide description of the features in Android mobile devices.

- Provide description of the cyber-physical social networks for mobile phones on Android platform
- Provide description of what tools are necessary to create mobile cyber-physical application for social networks.
- Explore topics related to both Android and physical social network.
- Provide some experiences and recommendations for developing mobile applications for physical online social networks.

1.3 Significance of the study

The thesis will be focusing on doing research in applications for Android mobile cell phones for cyber-physical social networks. In 2009 is expected that more than 60 percent of the worldwide population will have a cell phone. As more people acquire mobile devices, social networks that allow people to stay in touch with family and friends will be in greater demand.

This study will make a contribution to provide better understanding of the requirements to develop android applications for cyber-physical social networks. In this way, a developer can build applications that satisfy the requirements for successful products in mobile devices for online social networking. Moreover, this study presents to give recommendations to develop mobile applications for cyber-physical social networks and implications for future research.

CHAPTER 2

REVIEW OF LITERATURE

2.1 Android

2.2 What is Android?

Android is a mobile operating system based on Linux Kernel and developed by Google. To create applications using Android is necessary to use Java as a programming language and a SDK that can be downloaded in the Android developer site[5-7]. The first beta version of Android was released in 2007, within a few months over 1 million people downloaded the SDK. The term platform is used in Android to refer to the many components that Android provides including binaries, SDK, code libraries, and tools. The figure 1 shows the Android Emulator that is provided by the SDK to test applications.

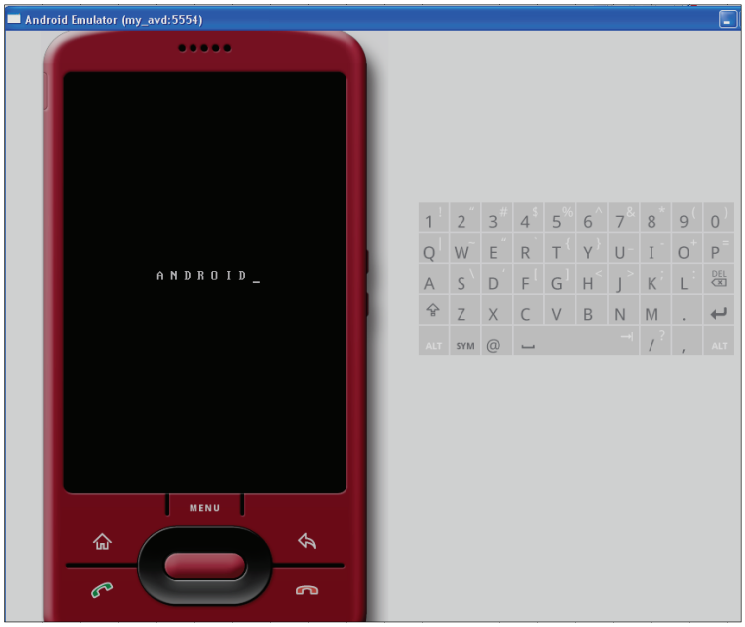

Figure 1. SDK Android Emulator

The Android project was created in 2007 by a group of forty-seven companies leaded by Google which formed the Open Handset Alliance. Some of the companies are Intel, HTC, Motorola and others. The main purpose was that each company provides help in developing open standards for mobile devices to accelerate innovation in mobile platforms[3, 8]. The idea was to enable developers with a set of tools and libraries to build applications with no restrictions to the phone features. For instance a developer has the rights to design an application using the features of the phone as the phone dialing, camera, and others. A total access to mobile phone capabilities makes appealing the use of Android to develop new products that offer new features and experiences to the users [7, 8].

Two different licenses were used to release Android. First, the kernel which is the main component of the operating system was released under the GPL license. On the other hand, the Android Platform was released under Apache Software License (ASL). These both licenses are open source. However, Android was released with two different licenses because Apache is considered less restrictive for commercial use. The Apache license allows developers to create applications and distributed with commercial purpose to obtain financial resources [9].

The Android platform was formed to offer alternatives to developers. Based on open source framework any third-party applications created using Android platform is treated as the same as the natives designed by mobile phone companies. It is a win-win situation for the developers who now can develop applications without acquiring any proprietary development tools. Moreover, developers can be certain that their applications will be treated equally with no preference because manufactures as well as independent developers use the sa[10]me API (Application Programming Interface) and executed in the same run time [5].

2.3 Benefits of working with Android

The main goal of the Open Handset Alliance is to provide the optimal tools to allow a quick innovation in mobile devices. Now, the developers in mobile phone programming can have better conditions, tools and the extra benefit of working in collaboration with Android developer communities to find solutions and exchange ideas. It is expected that Android will bring more opportunities to create new application for mobiles that allow flexibility and give a better experience for the cell phone users[9]. Some of the benefits that come from working on this platform are [7-10]:

- First, Android is open source which means the Android API is free for use to any developer who wants to create Android applications. This open source feature makes possible for talented people or companies to develop new and creative applications for the market. Another advantage is that SDK along with the Eclipse IDE can work in almost any operating system as Linux, Mac and Windows.

- Second, the application hierarchy allows to any application to be treated equally, no preference is given to any application, no matter if the application was developed by the manufactures. This is because third-party applications are treated equally because they were developed with the same tools and runtime.

- Third, Android platform contains a Software Development Kit that helps to developer to design and implement application. The Android SDK can be installed on one of the most popular IDE for programming Java Eclipse to help programmers to debug and polish their applications.

- Fourth, Android allows manufactures to personalize the platform by accessing all the features and tools of the platform. Therefore, they can release to the market new devices cheaper and quicker. Also, the consumer can enjoy mobiles easier to use and more affordable.

2.4 Requirements to work with Android

Android is not a programming language, the applications are created using Java in Eclipse Editor (not a requisite but highly recommended). The main tools available for developers are [5, 9]:

2.4.1 Eclipse Java Editor

Eclipse is an open source IDE (Integrated Development Environment). The main reasons that the Open Handset Alliance decided to recommend Eclipse to develop application in Android are [5, 9, 10]:

- Eclipse is available to download and use in any operating system platform including Linux, Mac and Windows. Moreover, Eclipse is considered to be easy to learn. The IDE is available on the Eclipse web page http://www.eclipse.org/downloads/. The latest version of the IDE is Galileo which works with the Android SDK. Figure 2 shows the environment in Eclipse Galileo with an Android Project:

- The alliance released an Android plug-in for Eclipse that helps developers to debug the applications easier making the learning process of creating application faster.

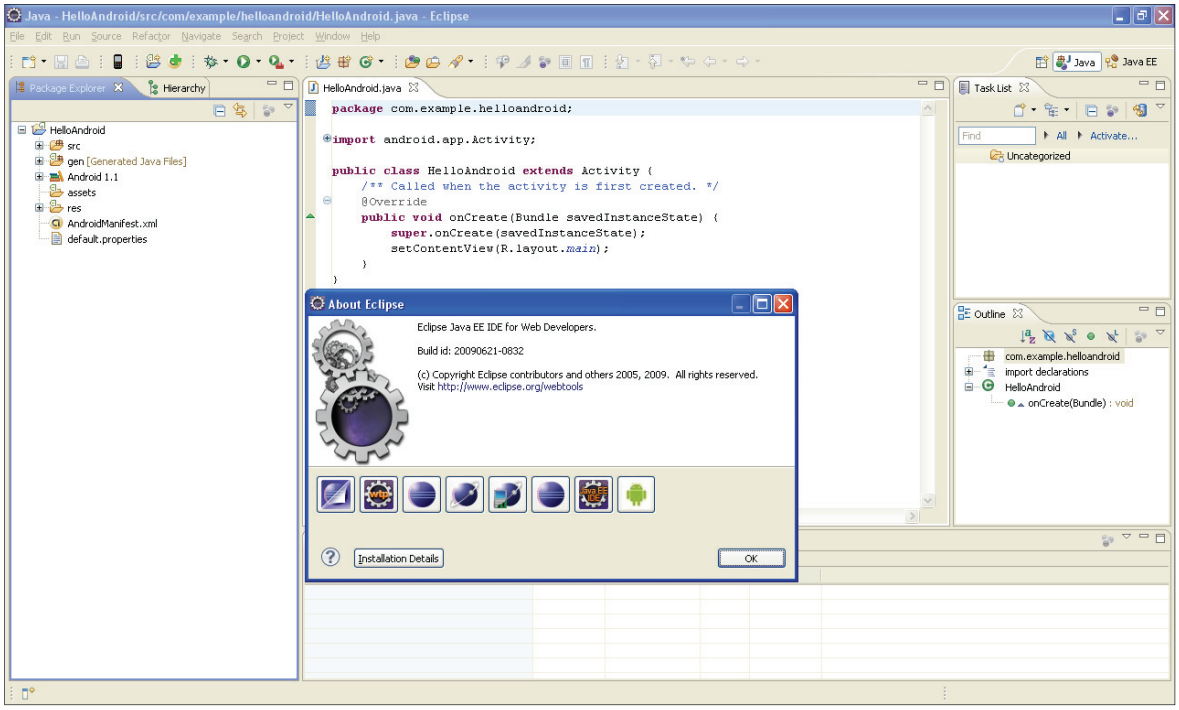

Figure 2. Eclipse Galileo Environment

2.4.2 Android SDK

Android SDK is a Software Development Kit that can be downloaded on the Android developer website (http://developer.android.com/). The zip package contains files, documentation, tools, APIS and samples to help the programmers to debug Android applications. The first stable release was 1.0 and the latest is the SDK 1.5 [5, 10].

One of the most important tools of the Android SDK is the emulator which allows the developer to simulate the applications without having to use the physical device. The emulation environment allows the programmer to debug, and compile an application faster simulating a real device according to application needs [5, 7, 11].

2.4.3 Java and Dalvik Virtual Machine

Android applications run inside of the Dalvik Build System relying on the Linux Kernel on file system management, process and memory [10]. Davilk Virtual Machine is used to run applications on Android. It was designed to avoid licensing issues and to work efficiently with the low memory and resources that are common with mobile programming. The Dalvik Virtual Machine is installed and configured along with SDK and plug-in. No extra work is required to be done for the developers on installing or configuring [9].

2.4.3 Eclipse Debugger

In order to facilitate the debugging process for the developers, Eclipse has a source level debugger that Android SDK connects with the Davilk Virtual Machine. No extra configuration from the developer is needed to start using this capability [10].

2.4.4 Logcat

When an application is being debugged is useful to have access to a log file with information about error messages from your application. Android provides this capability with the SDK. It is useful to filter the information to identify quicker the errors in the application. The command to obtain the log is the following: adb logcat [<option>] [<filter-specs>]. In Eclipse IDE can be access to go to Window->Show View->Error log [5, 10].

2.4.5 Android Debug Bridge (ADB)

It is a client server component created to provide a command line to administer the emulator. It also encloses three components: client, server and daemon. The adb is useful for tasks like starts and stops the server, uninstalls and installs applications or move files from or to the emulator. The adb comes with the Android SDK and can be access from any command line interface in Linux, Windows, or Mac[5, 7, 11].

2.5 System Requirements

According to the Android developer website (http://developer.android.com) Android is supported for the following Operating Systems:

- Windows XP (32-bit) or Vista (32- or 64-bit)
- Mac OS X 10.4.8 or later (x86 only)
- Linux (tested on Linux Ubuntu Dapper Drake)

Hardware minimum requirements are the same for running the operating system where the applications will be running. The Supported Development environments are the following:

- Eclipse IDE minimum version 3.3

- JDK 5 or JDK 6
- Android development plug-in.

Due to the advances of the technology it is recommended to check the requirements on the Android developer website http://developer.android.com/

2.6 Features in Android

Some of the most important features that come on Android SDK will allow programmers to create applications that give users new products that can give resourceful options to users. Android supports the reusability and replacement of the components. The following are features for applications that can be access on the SDK [7, 10]:

- No fees in licensing, distribution or development fees.
- Wi-Fi access, Bluetooth, 3G and EDGE.
- Access to GPS for location based applications
- Hardware control that includes features as using camera, recording, networks, microphone and others. Also, contains media libraries to work with a wide variety of video, image or audio file formats.
- SQLite database to be used for those applications in need of data storage.
- Camera, GPS, compass and accelerometer.
- Background services to support applications to run in the background.
- Native Google maps and location based services.
- Open source WebKit-based browser that allows access to tools for browsing the web.
- Support for 2D and 3D optimized graphics.
- Java Based application framework.

2.7 System Architecture in Android

Android is a software stack designed to work with mobile devices. The Android Architecture is designed to emphasize replacement and reusability of the components in order to support new programs to be written quickly. The components that interact in low level are written in C and C++. However, the user applications are written using Java. Android is composed for the following components: applications, application framework, libraries, android runtime, and the Linux Kernel[7, 9], [10] .

Figure 3 shows the components and subcomponents of the Android Architecture. It is important to mention that colors have special meaning: green is to remark that the components were written using C/C++ and blue represents that components were written using Java.

Figure 3. Android System Architecture.
"What it is Android" 2009. Android Developer http://developer.android.com/guide/basics/what-is-android.html

2.7.1 Linux Kernel

Android is based on the version of Linux 2.6 and relies on it to administer functions as security, memory, power and process management, driver model, and network stack. The Linux kernel allows developers to take advantage of the hardware features and capabilities of the Android phones. It works as an abstraction layer for both hardware and the software stack[6, 7, 10].

2.7.2 Libraries

Android libraries are a set of subroutines written in C/C++ that contain the code and the data necessary to provide services to Android applications. The native libraries in Android platform

represent the core power of the Android Platform. Some of the most important libraries are [6, 7, 10]:

- Media Library to support audio, images and video media. Some of the formats accepted are: MPEG4, H.264, MP3, AAC, AMR, JPG, and PNG.
- Surface manager: administer the display management.
- Graphic libraries to support SGL, OpenGL for 3D and 2D graphics
- SQL for applications in need of data storage and retrieval.
- System C library to be used in embedded Linux system.
- LibWebCore and SSL to be used in web browsing and security over the internet.

2.7.3 Android Runtime

Android runtime consist of two main components, the core libraries and Davilk Virtual Machines that incorporate most of the functions available in the libraries of the Java programming. In Android, each application is assigned its own process and instance of the Davilk Machine [7, 12]. The Davilk Machine was written to support the efficient running of multiple VMs on the device. More detailed information about the two components is bellow provided [6, 7, 9, 10].

2.7.3.1 Davilk Virtual Machine

The Davilk Virtual Machine was created to run applications fast in an environment by nature limited by memory, space, CPU, and battery power. The Davilk Virtual is used as a middle level to guarantee that android developers do not have to deal with the hardware implementation. This virtual machine creates from Java applications, an executable and optimized file with dex as file

extension. To obtain Java translation to dex file, the Davilk VM uses the tools integrated in the Android SDK[6, 7].

2.7.3.2 Core libraries

Core libraries are programming applications in Android are written in Java. However, Dalvik Virtual Machine is not Java Virtual Machine. The main function of these libraries is to provide most of the functionality of the core libraries in a Java environment [6, 7].

2.7.4 Application Framework

The application framework provides classes that are written in Java. The classes are used to create Android applications and handle the application resources and user interface. All the applications, both native and third-party, have to use this framework. Also, the application framework provides a collection of services and system that include [6, 7]:

- A suite of views to write applications and include elements as grids, texts, buttons, and browser for embedded devices
- A content provider to enable the data sharing or data accessing from another applications.
- A resource manager to allow the access to resources as graphics or strings.
- A notification manager to show customized alerts in the status bar.
- An activity manager is used to handle the application lifecycle.

2.7.5 Application Layer

The final layer is the application layer where the native and third-party applications are written in Java using the same API libraries. It includes a set of core applications such as calendar, maps, browser and others [6, 7].

2.8 Android Application Life Cycle

It is vital for developers before creating applications to understand the life cycle of Android applications to start creating them. A life cycle is a set of organized steps that allow an application to have a start and termination. In Android, as any other applications must have a life cycle to determine a beginning and finish point to ensure consistency [5-7].

An application in Android is written in Java. Each application is isolated from the others; it has its own virtual machine, Linux process, user ID, and consists of the following components that are essential to build a project: activities, services, content providers, intents, broadcast receivers, and notifications [6].

2.8.1 Activities

An application might consist of one or several activities depending on the goal. It is not necessary for an application to have a user interface. However, if an application needs interaction with the user it will have at least one activity. It is the user through the interface who triggers the events. An example of activity in android is a user choosing in a list or menu an option, then the recorders starts working because of the user actions [6, 7]. The following picture represents the life cycle of an activity:

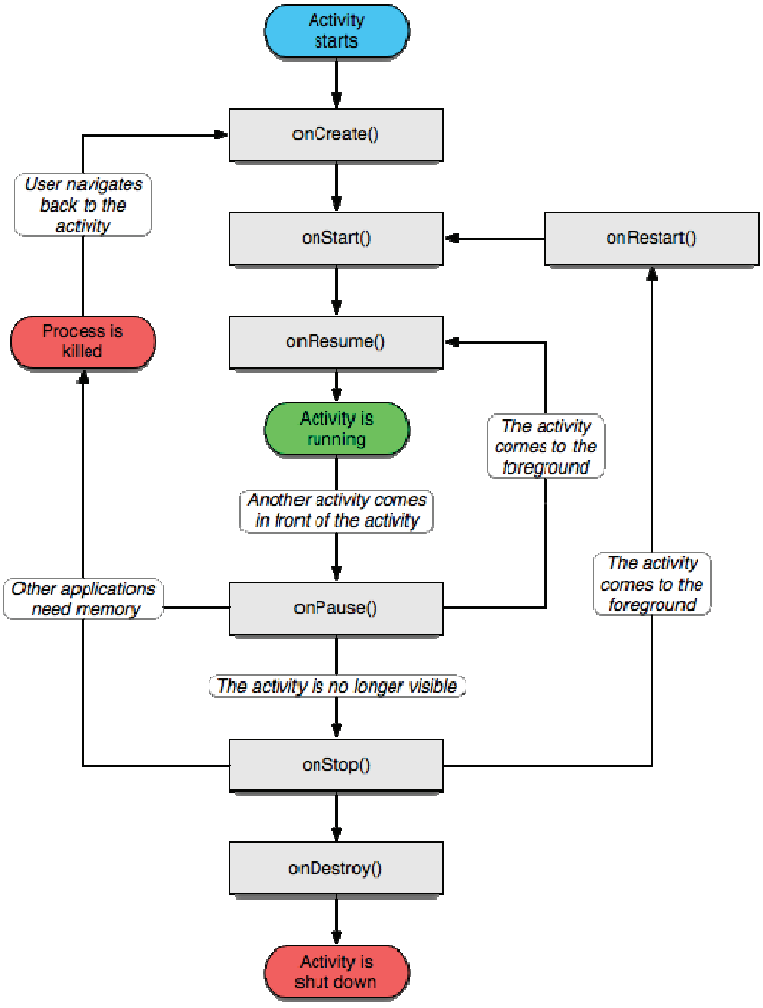

Figure 4. Android Life Activity Cycle "Activity | Android Developers," http://developer.android.com/reference/android/app/Activity.html.

An activity stack is used to handle activities. If a new activity starts it is located on the top of stack. This new activity is now the running activity until a new one is placed on the top the stack. There are four states in an activity life and are represented in colors in the above figure [7]:

- An activity is considered active or running when is on the top of the stack. It is considered to be triggered by user events and interacting with the user [6, 7] .

- An activity is considered be visible or paused when is not responding to any user events. While it is visible it holds state, member information and it is bounded to the window manager. In case the system needs more resources, a visible application can be killed to obtain them [6, 7].
- An activity that is obscure or stopped means that is hiding from the user. However, an obscure activity still holds state and member information. This type of activity has a high rate to be killed for the system in case additional resources are needed [6, 7].
- Activity paused or stopped is when the system release resources in memory by either asking to terminate or kill the activity[6, 7].

The figure 4 displays the three key loops to supervise an activity to help enforce consistency: the entire, visible, and foreground lifetime. The rectangles in gray in the figure represent the events that those states launch [6].

- The entire lifetime of an activity occurs when onCreate(Blunded) was called until the event onDestroy() is invoked releasing the resources that the activity was using. The event onCreate is called just once in the activity lifetime[6, 7].
- The visible lifetime occurs when onStart() is invoked until onStop() is called . The OnStart() event is used when the activity has interaction or need to be displayed to the user. The resources needed are hold to present the activity to the users [6, 7].
- The foreground lifetime means the *activity* time to interact with the user. It occurs when method onResume() is called until onPause() is invoked. If the event onPause() is called, the activity will be paused and it exist the possibility that it can be destroyed. If the activity is not killed it can be called to be the runtime activity using the onResume() event [6, 7].

The following table shows the movement through activities. This table can be found in the Developers Android website.

Table 1. Activity life cycle movements. "Activity | Android Developers" http://developer.android.com/reference/android/app/Activity.html.

Method	Description	Killable	Next
onCreate()	It is the first method to be called in the life of an activity. It corresponds to the place where a developer creates the views, customize list, etc. It is always followed by onStart()	No	onStart()
onRestart()	When an activity has been stopped before it starts is necessary to call this method. It is always followed by onStart()	No	onStart()
onStart()	This method is called when an activity needs to interact and be visible to the user. Depending on the needs of the activity it can be followed by onStop()or onResume()	No	onResume() or onStop()
onResume()	This method is called when the activity will start user interaction. At this moment is located on the top of the stack.	Yes	onPause()

Method	Description	Killable	Next
onPause()	This method is called when it is necessary to resume another activity. The activity no longer has access to the screen. It was designed to save resource as battery, memory. The system has the option in case of low resources to kill it.	Yes	onResume() or onStop()
onStop()	When it is called the activity is no longer visible, another activity has taken the foreground or the activity is being destroyed.	Yes	onRestart() or onDestroy()
onDestroy()	The final method in the activity life and it represents the end or termination of the activity. It can be triggered by the system when it needs to obtain more resources.	Yes	Nothing

2.8.2 Services

The service runs on the background with no time limit and lacks of a user interface. They are similar to the daemons on the operating system. The most common example of services is the MP3 player that plays songs from a list, when a song is finish, the service continue with the next song on the list[7, 10].

2.8.3 Content providers

A content provider is the only technique provider by Android for applications in need to store and retrieve data. Also, it permits other applications to read or write data by allowing access. Android gives the developers the option to create content providers to make their own data public[7, 10].

2.8.4 Intents

Intent is an asynchronous message used to activate services, activities and broadcast receiver. Typically, it is used to jump from one screen to another and describes what the application intents to do. The structure of intent is composed by the action that wants to be performed and the data to be subject to change [5-7].

2.8.5 Broadcast receivers

A broadcast receiver is a component that deals with intents. It is only job is to receive messages of an event and start application to answer the request of an intent. The messages can also come from the system to notify for instance that the battery is low. It is important for the broadcast receiver to distinguish them and filter to take the appropriate actions [6, 7, 10].

2.9 Application priority in Android

The system has the power to kill processes to obtain more resources. To make a decision, the system has to distinguish the importance rate of the processes. Figure 5 shows the priority that is assigned to processes [6, 7].

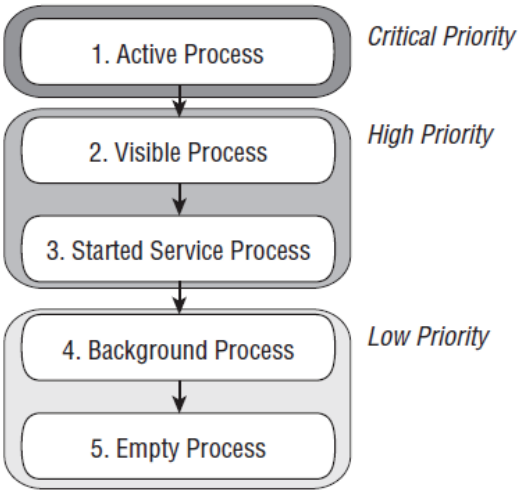

Figure 5. Priority tree application termination. Source: Meier,2009

Android applications will be running and not be terminated unless the system needs to obtain more resources. When the system needs to release memory but two applications have equal priority. How the system makes the best choice? To solve the conflict, the system will choose to kill the process that has held the longest lower priority [6].

An *active or foreground process* has a critical priority. Rarely, the system will kill a foreground process and not many of them exist. The system will only terminate an active process circumstances as very low memory where not all the process can be running. A process to be considered as a foreground or active must meet some the following criteria: [6, 7].

- The user is interacting with an activity that the process is holding and onResume() method has been called.
- It include a service object that is running the following methods: onCreate(), onStart(), onDestroy() .
- It has a BroadcastReceiver object that is running on its onReceive().

Visible process is a process that includes an activity visible on the screen but it is not foreground or interacting with the user (its onPause() method has been invoked). It is considered to be a high priority and it will not be eliminated unless it has to be killed to preserve the active foreground processes [7].

Started service process is a process that it does not interact with the user directly and it has been called by the starService() method. It will not be eliminated unless the system needs more resources. A classical example of this type of service is when the user is downloading data or running the mp3 player [6, 7].

Background process is a process that includes at least one activity that is not visible to the user (its method onStop() has been called). The system at any time can kill this process to obtain more resources and give priority to the three previous processes described. It is common to have many background processes running. The system will maintain a LRU (Last Recently Used) list of background process to guarantee that the last one used will be the last one to be eliminated [6, 7].

Empty process is a process that is used as a cache to improve the performance of the system. The system will be eliminating this type of processes frequently to keep a balance of the processes cache and kernel system cache [6, 7].

A process priority might increase if it is serving other processes. For instance, process A that is serving a process B will always have a higher status because process B depends on A to continue running [7].

2.10 Create applications in Android

Before creating Android applications the Eclipse environment, Android SDK, and Sun's Java Development kit should be configured. There are several websites that have well-written tutorials to configure Eclipse and Android. However, the developer Android website is highly recommended. It contains detailed explanations and recommendations according to the operating system where Android will be installed. The URL is: http://developer.android.com/sdk/1.5_r2/installing.html[7].

A developer must have programming knowledge in Java and XML to understand and write Android applications. This section will explain how to create an application in Android, the classic example "Hello World" will be explained at some detail [7].

Before running the emulator, it is necessary to create an AVD (Android Virtual Device) which is use to define the device settings and system image. Follow the next steps [7]:

- Open a command prompt (cmd) or a terminal in Linux
- Go to the directory where the Android SDK is installed and go to /tools and run the following command: **android create avd --target 2 --name my_avd**
- The command will generate a message asking to create a hardware profile. Usually, "no" is the standard answer; unless an Android developer needs a more customized AVD.

Now, the following instruction should be followed on Eclipse Editor integrated with Android [7]:

- Go to menu File>New>Project. It should be displaying the option create Android Project in the wizard
- Select Android Project and click Next.

Write the project details (Figure 6) with values:

- *Project name: HelloAndroid* → represents the folder that will have the project files .
- *Application name: Hello, Android*→ the name of the application, it will show on the device
- *Package name: com.example.helloandroid*→ the package is named according to Java rules, when creating application, the developers need to follow the standard and name the package according to their institutions', organization's names.
- *Create activity: HelloAndroid* →it will represents the activity class in Android. It is a classical class and is optional.
- *Min SDK Version 2*→ It represents the API level to be used in the developer application. Each newer version contains API that has more functionalities and option for the programmers.

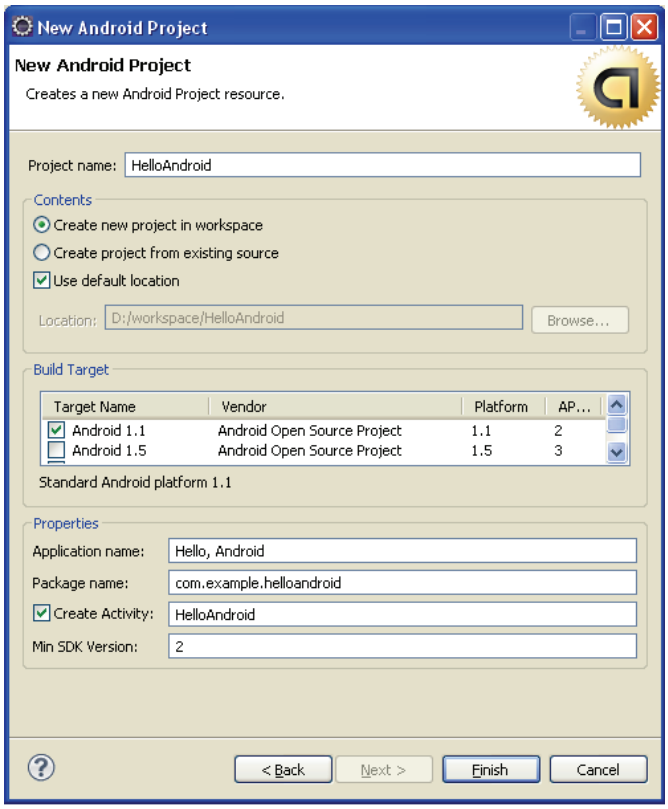

Figure 6. Creating a new Android project

- Click on Finish

The code in HelloAndroid.java in the Eclipse Editor looks like Figure 7. Activity is use as the base class. Each Android application needs to have at least one activity. Visual component in Android are named *Views* and it is not included on the code in Figure 7.

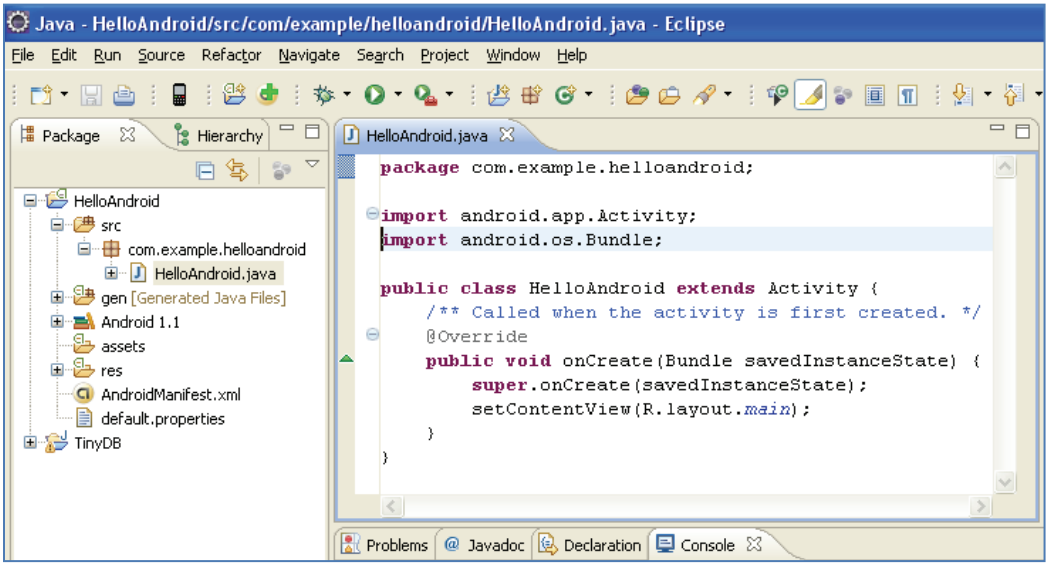

Figure 7. helloandroid code application in Eclipse Project

To add a User Interface (UI) adding the following code displayed in Figure 8 will do the trick. If the setContentView() method is not called, it will not display the UI [7]:

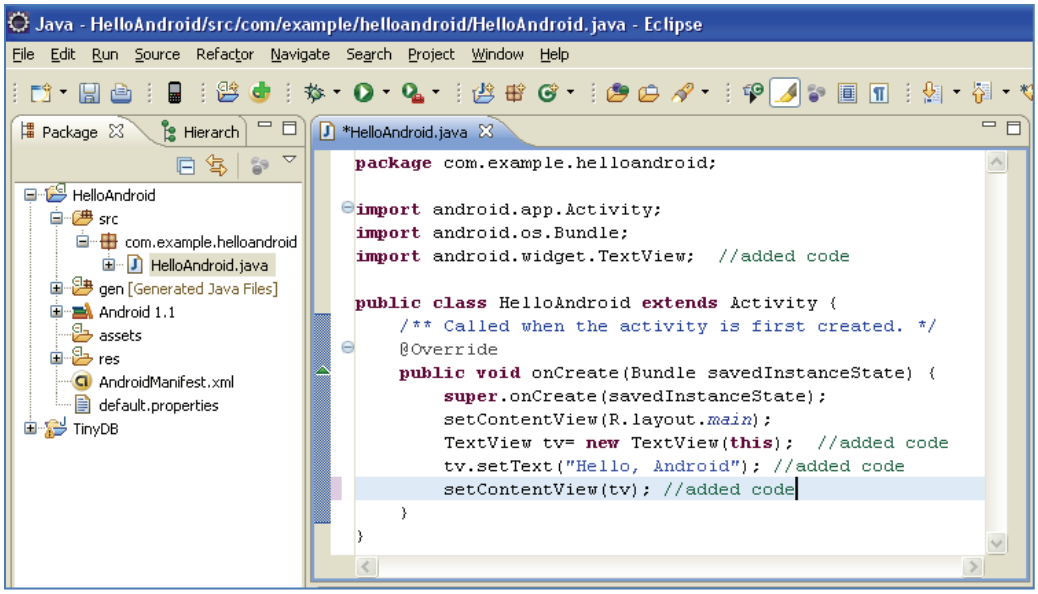

Figure 8. Adding a UI to the helloandroid application.

To run the application on the emulator choose run → run and select Android application, this will run the emulator to display the application. Figure 9 displayed the result[7].

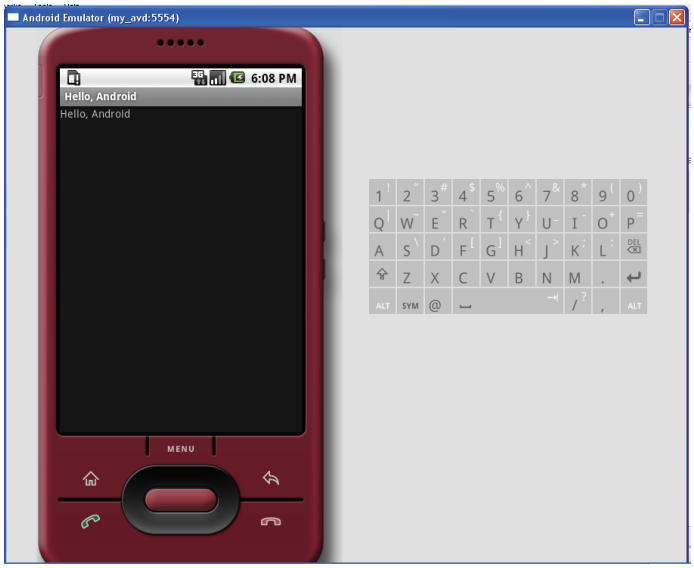

Figure 9. Running on the emulator the androidhello application

2.11 The AndroidManifest.xml File

The AndroidManifest.xml file is a required file used to control the global settings in an Android application which should be located in the application directory. It contains information about how to handle processes, activities, permissions, components, linked libraries, capabilities, minimum level of the API requirements, and other necessary information to interact with the Android system[6, 7, 10].

2.12 Android phones to test applications

Before releasing new applications to the users, developers should test them on real phones to ensure the quality of the application. Usually, developers prefer to work on unlocked phones (no contract attached to any particular provider). The following are some choices on phones that developers can obtain in order to test that the application runs correctly [7]:

2.12.1 T-Mobile G1

The T-mobile G1 is called the Google phone and is a good option for testing applications. It allows a developer to install the applications and be able to test them using real hardware [7].

2.12.1Android Dev Phone 1

The Android Developer phone was designed to be SIM-unlocked and hardware-unlocked without being attached to any provider. Because, it is intended for developing use and any developer who does not have access to T-Mobile networks should consider this option. However, to buy this phone, it is necessary to complete a registration on the Market site, pay a fee, and buy the phone. In June 2009, the cost of the phone is around $399 (American dollars)[7].

2.13 Social Computing

Social computing is not a new concept. It was mentioned in 1945 for Vannevar Bus. In his paper, he introduced a device named Memex, and also discussed ideas as groupware and computer collaborative work. The first social networks targeted services and solutions related to interface, group collaboration, and communication. Nowadays, the spectrum of the social computing has grown [13]. One definition to explain the new areas where social computing has found a place for expansion as business, public sector, online communities and others [13].

Figure 10 shows the infrastructure of the social computing. The infrastructure is based in both social and computational sciences influencing each other. Social sciences have concepts in diverse areas as psychology, anthropology, economic and others that combined with computational theories provide a base for the development of social systems. The major contribution of computational sciences is to offer the technological support by providing databases, multimedia, wireless, and software engineering to help build these simulated societies. Social computing applications are developed according to particular requirements. One of those requirements is to create or improve software that offers better techniques to enhance communications and interaction among individuals or groups [13].

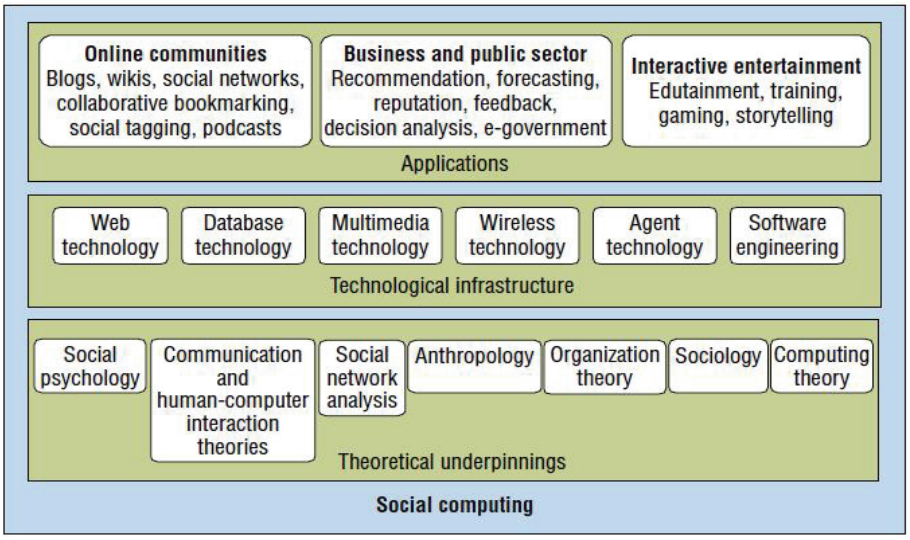

Figure 10. The infrastructure, and applications of social computing. Source : F, Wang et al. Social Computing: from social informatics to Social Intelligence.2008. IEEE

2.14 What is a social network site?

Nowadays is common that people are using at least one social network to maintaining communication among friends, families or coworkers. As a result many companies are developing products to take advantage of the opportunities that social technology has opened [14]. A *social network* is a network, a complex network that involves people connections (nodes) by some sort of relationship as friendship, family, relatives or interest, recommendations, conflict, etc (edges). A social network is not a new topic. It has been studied for at least 50 years. Nowadays thanks to the technology advances and the Internet, the interest in social network has been stimulated by its potential to make connections around the world. A few excellent representations of large social networks that generate income by offering publicity or services are: Facebook, Digg, Twitter, Facebook and others [15].

The Social networking site (SNS) has allowed since its beginning the existence of virtual communities that promoted the participation, communication and interaction of its members. One

of the first SNS that looked similar to the modern SNS was introduced in 1997 and named six degree. It combined features and ideas that other sites had. For instance, AIM and ICQ supported list of friends, Classmates made available to users to join groups according to high schools or colleges. Six degrees started as the first site offering all options in one site. However, the site closed in 2000 because it did not find a way to be profitable[16].

In years 2002 and 2003, some SNS became more popular. Then, some large companies as Google or Microsoft tried to conquer people's attention with their own SNS. Google launched Orkut which was very popular in Brazil but not in US. On the other hand, Microsoft had the Microsoft Live Spaces becoming well-accepted in other places but not US. In 2003, MySpace was the favorite in US and it allowed users to customize their page profile. It grew rapidly, and eventually changed its policies to allow minors to have sites which made MySpace very popular with teenagers [16].

Facebook was born in 2004. First, it was a social network designed just for college students. Later, it was extended for high schools. Then, it was open to everybody to sign in an account. Currently, in many colleges, Facebook has 80 percent presence in undergraduate community. One of the main reasons of its popularity is the features that other SNS did not provide. A good example is to show the user contact information[17].

Lately, new SNS have become more popular. Some of them focus on targeting specific groups to captivate certain audience. For instance, MyChurch.com is designed for Baptist Christian people to provide information about how to meet Christians singles, church address, etc. Most SNS are intended to be centralized around people, conceptualized to be personal, each user is the focus on his/her own community[16].

Recently a report by Joint Research Center European Commission (JRC) shows a significant increase since 2003, in the use of social applications including social network sites. It informs than 1 million photos are uploading every day, new objects are created in numbers of billions. In Europe, JRC estimates than more than 20% of Internet users have the habit of interacting with applications in social networking, social gaming and tagging. Another interesting statistics data shows the usage in social computing for continents or countries. Asia is the leader 50%, United States 30%, and Europe near 20% of Internet users using social computing applications [18].

2.15 Benefits in using social networks

Why social network is so important? A social network provides a way to satisfy the basic need to be informed. It is vital to people to feel updated and communicated with the links that are important to them. The usual links in SNS are composed of coworkers, family members, friends or particular interests. Both social network sites and people obtain benefits, SNS can make profit by offering publicity and services. On the other hand, users obtain accurate information about their personal interest Moreover, thanks to this type of technology, limitation as geographical distance has been reduced allowing people to establish connections with others around the world [14].

To companies socials networks are appealing to reach users who are likely to accept their products publicity or services. With social networks is possible to filter users according to specific data and find potential customers. Some of the tactics used for companies to catch the attention of users are: promoting discussions, share experiences, and ask for feedback to their products or services to improve them [19].

2.16 Basic features of a social network site

The success of a social network is commonly based on the features that can provide to make people interest in engaging, using and inviting others to use the SNS. The common options that are displayed on social networks are[14]:

2.16.1 Accounts

It is believed that large amount of accounts is equal to have a successful SNS. Users create their accounts in the SNS in order to register and obtain an identity. Commonly, the SNS implement systems to verify the authenticity of the information entered by the users. Some of the measures to increase the security and the reliability of the data are: e-mail confirmation, password encryption techniques, and CAPTCHA system (Completely Automated Public Turing test to tell Computers and Humans Apart) used to make sure that who is signing for an account is human and not a computer used with purposes of spam. Also, in this process of creating accounts the policies agreements are informed to the users and requested to be accepted [14].

2.16.2 Profiles

A profile is considered an additional feature of the user account. The distinction is that the account has the login information. On the other hand, a profile includes the user's personal information like his/her interest, background data, hobbies, occupation and others. It is important that the SNS offers to users a way to control the information that will be displayed to respect privacy [14].

2.16.3 Friends

Friends feature is considered one of the main reasons for people using SNS. A friend is another user who has been authorized to see information, and at the same time is willing to share it back. A friend is considered to have some sort of connection with the user as relative, colleague, acquaintances or people who shares interest with. The key of success of SNS is to provide users tools that provide a way to search for friends, invite friends to use the SNS, suggest new friends, alerts system that can provide information about the activities of the connections [14].

2.16.4Messaging

In a SNS the messaging is a vital feature that provides to users a way to communicate with each other. There are different methods for guarantee messaging exchange: a message sends it by email through the site, an email application, chat, alert service and others [14].

2.16.5 Media galleries

Media galleries provide a media system to users to share and store photos, videos or any files. By using the gallery users can use, display, or download from their page the files that they are interesting in sharing. Some basic options for media galleries are: one or multi files upload, image edition, resize of images, maximum size, and create folders[14].

2.16.6 Blogging

People in a community seek ways to publish and share thoughts, comments, and ideas on a particular topic. A blog offers to users the mechanisms to post articles, links, videos and others to express feelings or attitudes toward an issue. When someone posts a new comment, the SNS has

an alert system to inform to the participants on the discussion that additional information has been posted on the topic. By doing this, encourage users to continue exchanging communication [14].

2.16.7 Message boards

It is a web application known also as forum and it is used for the users to discuss about a particular topic. One feature is that all the opinions are not usually eliminated and can be seen at any time. It has the purpose to encourage discussions and sharing ideas to solve a particular issue. A message board is composed for categories to identify the topics. A category might contain many forums according to its needs to complement the topic[14].

2.16.8 Groups

It is composed of people that are grouped toward a topic appealing to them. It exist two types of groups; public and membership. A public group anyone is free to join it. On the other hand, a membership group has to wait for an approval to join that group. A group usually contains itself features as forums, blogs to be used for the members to stay in touch and express opinions toward the particular topic which the group was created[14].

2.16.9 Comments and tags

Comment is another feature to provide users a method to express their viewpoints of the content posted for other members. A SNS implements the commenting as a rule because it is useful to write opinions about photos, profiles, forums of the other members or themselves. Moreover, other features have commenting as common denominator, the purpose again is to encourage interaction among the members[14].

Tag is a word associated with particular objects that represent text, images, audio, video and others. It can be linked to one o various object in the community by one o more users. It purpose is to encourage the clicking around the object and provides more motion in the community. Tagging is useful to identify fast objects through keywords that are assigned to them. Figure 11 shows an example of tagging and comment on a figure [14].

Figure 11. Tags and comment example. Source: Siemer, A. p 411 (2008)

2.17 Mobile Social Computing

Mobile Social Computing is a natural step in the advance of social computing. Social and mobile features are being unified. Now, they are found in forms of mobile blogging, mobile podcasting and mobile social networking. Their applications and services have increased in use [18].

Mobility brings a new aspect to consider in the mobile context Figure 12. **Mobile Context.** The mobile context is defined by Ortiz as "*the set of and the intersection between facts, events, circumstances, and information that surrounds the (mobile) user at a given point in time"*. The mobile context elements and interactions are represented in Figure 12. The social context plays an

important role because it represents the links, properties, and actions of the person's social context. Some examples of the social context are: friends, coworkers, calendar, location, actions as invitation to social events, etc. The user experience is enhanced with benefits as: precise information, time, connection, and others [20].

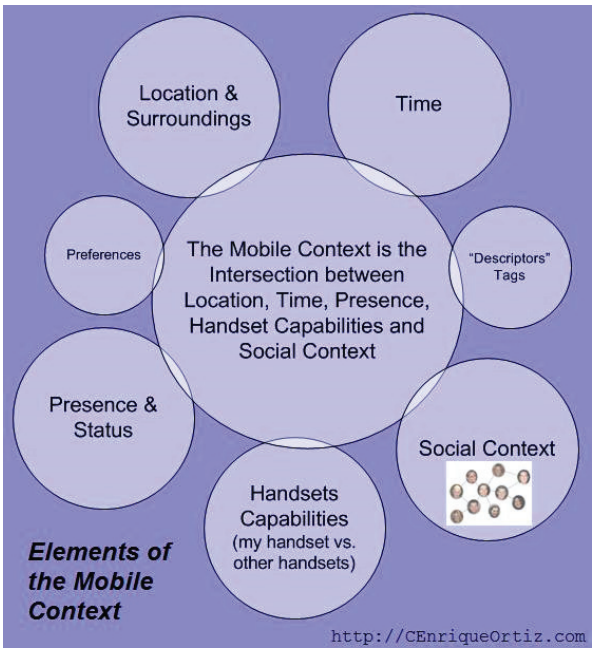

Figure 12. Mobile Context. Source: E. Ortiz 2008

In Figure 12, the middle circle is also composed of the platform where developers create applications according to carriers and mobile device needs. Some popular platforms are Java ME, Symbian OS, Mobile web, Windows mobile and others. Developers need to keep in mind that success depend on providing to users a good experience [20].

2.18 Mobile Social applications

In the mobile social computing field there has been a great deal of progress. New terms are emerging like mobile social stream, mobile social casting and others to enhance new ways for people to stay connected and informed [18].

2.18.1 Mobile micro-blogging

Micro-blogging is a collaboration tool that consists in a micro version of blogging; it is also supported for the use of mobile phones. It allows easy and quick exchange of messages and to post users status. Twitter is one popular service in micro-blogging. It allows users to exchange messages through tweets (messages that appear in reverse date, the latest date is posted first). Tweets are sent through different ways as text messages (SMS), external applications, etc [21].

2.18.2 Mobile lifestreamming

It is a version of micro-blogging. However, it includes adding not just text but media files as audio, video. It has the purpose of maintaining a history of user activities. Some of the popular services in lifestreaming are Tumblr, Jaiku and Lifeestrea.ms. This type of service was created in 2007 and it is available for mobile devices adding permanent connectivity to users [18].

2.18.3 Mobile social tagging

A tag is associated with a keyword that was introduced by the user. In the mobile social tagging this is taking to another level by using the integrated camera. For instance, when a photo is taken suggestion are displayed for tagging the photos before publishing. This application is offering by ZoneTag (http://zonetag.research.yahoo.com) [18].

2.18.4 Mobile podcasting

This term has been used in the mobile industry to describe the action of a new mobile handset option to find, subscribe and download podcast made by third parties [18]. It uses RSS 2.0 to allow subscriptions. Podcasting enables independent third-parties to create new media content that can publish easily. An application will be in charge of informing the user that new content has been added and downloaded. This type of delivery method is becoming popular not only for MP3 players but also for mobile devices[22].

2.18.5 Mobile social network

Many companies are interesting working on duplicate the success of the SNS for mobile devices. Some famous SNS companies are MySpace and Facebook. Also, another emerging business is creating new mobile SNS as Migg33, ZYB and Mocospace. These new type of SNS are offering features oriented to location detection and enhance the iteration of users which are projected to appeal them. Also, the latest smart phones have integrated sensors which can be used for creating applications that have to deal with user location and nearness[18].

2.18.6 Mobile social gaming

Mobile social gaming has been restricted due to technology and cost. It is still a debate if there is a good option to offer virtual world in handsets. There are carriers and companies that are developing options in mobile social gaming. For instance, Google is developing a 3D virtual world using the Android platform [18].

2.19 Android and social networks

One of the big expectations with Android is that it will encourage the design and development of the mobile applications intended for social networking. Currently, in Android a developer can create such applications because it provides the capabilities to facilitate the development process. It is expected that mobile applications for social networks will increase in popularity. Because of their simple use and the potential of allowing users to be connected all the time with their important contacts by sharing ideas, files, or even videos [10].

2.20 Cyber-Physical Systems definition

Cyber-physical system consists of set of elements that combine physical-world and human interaction[23]. A cyber-physical system would integrate the features of both physical system and computational system to provide solutions that cannot be resolve by using just one of them [24-26] . Also, a CPS is composed of physical elements which are distributed within an area. The components could be either fixed or mobile depending on the needs. Some communication channels must be active to provide transmission of the information. The sensors will provide information and depending on it, the physical world will act accordingly [27].

Some implicational examples for CPS are those dedicated to electric grid management, transportation management, factory automation, military applications, embedded system and others [27]. It is believed that CPS application will be the next step in IT development causing great challenges in medical devices, car systems, conservation of energy, water resources, telemedicine, and manufacturing. One of the best implicational examples in CPS is one project conducted by MIT where robots are being taught to take care of garden to produce cherry,

tomatoes and other goods. The goal of the project in an economical viewpoint is to find a way to minimize the manual labor to cultivate plants that require intensive care and work [28].

However, CPS applications still need to get more mature. It is necessary to raise the level of confidence before starting them to use commonly. This because CPS's work with delicate variables in traffic systems, medicine instruments and others that require a high level of precision that current methods still not provide [29].

CPS has challenges to face and the application designs are difficult. Most techniques and software technologies are related to work with systems that do not have to deal with physical variables and circumstances [30]. In a common system it is essential that reliability and predictability to be a part of design and deployment. In CPS these requirements are expected even more because applications are designed for sensitive systems as traffic control, health care, safety systems, and others that work 24x7. However, obtaining predictability in physical world is a hard problem and these applications have to deal with unexpected events and respond appropriately. Lee in his paper suggests a solution which is to use predictable concurrent computations whenever possible and find deterministic solutions as much as we can [30].

2.21 CPS design models

To develop CPS it is necessary to consider other variables into account like security, tolerance and others. One model proposed by Karsai and Sztipnanovits suggest us to think about the elements that play part in the design of the CPS. They proposed a model to be used as a instantiation to understand better roles among the elements [24].Figure 13, shows their vision and interaction where "*model world*" refers to the elements to be implemented in the "*real-world*" which is on the right. The application software takes part in the real world solution and includes

operating system, engines and networks that are necessary to establish the link to work with the physical system[24].

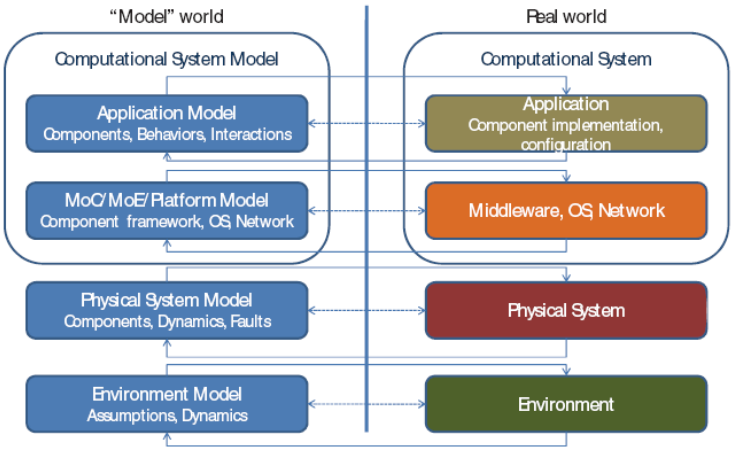

Figure 13. Models and the real world in Cyber-physical Systems
source: G. Karsai, and J. Sztipnanovits.

2.22 Challenges in CPS

CPS developers encounter a wide variety of challenges; many of them come from those in embedded system. Some of the major problems are: model integration, system integration, support for certification, model changes and fault management [24].

One difficult problem is the *model integration* that arises when it is necessary to simulate the whole CPS. There are some tools that can provide help, for example Simulink that can reproduce simulation in continuous time. The other model used is *High Level Architecture* (HLA). In HLA it is necessary to have a coordinator. This will help to establish a communication link with different computer simulations regardless of the platforms that we use to create them. The employment of different types of models and tools require some level of abstraction that makes the solutions and the verifications more complicated in the development of CPS [24] .

System integration is considered to be the most difficult feature of CPS engineering and also it is where models can be of more importance because the physical and software components in CPS must be analyzed, designed and modeled together. Karsai and Sztipnanovits created an integrated and simulation development model where parts are replaced by real implementations piece by piece gradually[24].

According to Karsai and Sztipnanovits, *integration* should be the core element in the design of the cyber-physical system. They proposed a model where continuous integration is needed with assumptions about how the system should be built in layers displayed Figure 14. The main layers are application, platform, physical system and finally the environment. The computational platform layer interacts with the physical systems through sensors and actuator. On the other hand, the communications between the platforms are linked using API's.

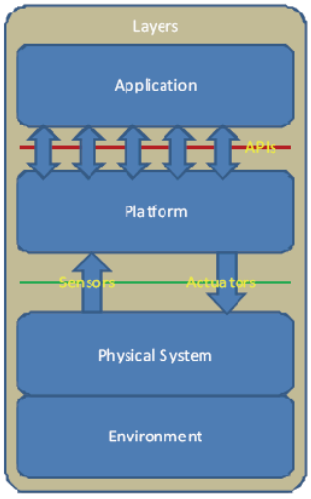

Figure 14. Layers of CPS.
Source: G. Karsai, and J. Sztipnanovits,

Another hard problem for CPS is its certification. The certifications of critical systems like airspace, healthcare, and traffic control systems are not an easy task. There are three methods

used to certificate: simulation-based, verification-based, and hardware-based testing. In the first method the environment and the physical system are simulated with high-precision in details. In verification based core consists of examination of the code and the system model. Third, the hardware-based testing works with exhaustively testing the hardware to be used in the CPS [24].

Other group of challenges includes management of faults. It is a hard problem to handle unpredicted events in a physical system. As any other software, a CPS needs to be prepared for handling the unexpected problems and to obtain the appropriate results. To design a CPS a common process to use is "*Fault Detection Isolation and Recovery*" FDIR. Following this process provides a way to detect faults in order to isolated and apply the corrections to the problem[24]. However, CPS's are difficult to control in every aspect because it is hard to prevent everything in the environment, for instance rain might affect the communication between the devices or there are some factors like noise interference that affect the quality of the spectrum[31].

Everyone while designing the system needs to consider Quality-of-Service (QoS), which includes good performance and quick execution. However, in cyber-physical systems this issue is different, it is more important to have actions that are repeatable and constant rather than taking less time. For instance, when a car user wants to start engine, the best response is not to do it early, the best time is to start when is expected to do[29].

Another challenge for CPS is the respect of the user's privacy. It is vital to build system that will resist attackers from obtaining sensitive data without having to sacrifice the quality of service. This is not an easy task; some people would like to choose a better quality of service and goods rather than a high protection of the data. This customizable behavior will depend on the user context [31].

2.23 Physical Topology

Currently, most of the cyber-physical system designed and running use sensors or actuators, they are not great in complexity or size. It is expected that those CPS will give the first perspective of effectiveness and functionality where networking will have a special role to play as integrating all the elements in the CPS[25].

Xia and Ma have an interesting vision of the future physical topology for CPS where they take in considerations the role of sensors, applications and the networking Figure 15. In their paper they mention that a CPS system could consist in several subsystems interrelated with wireless sensor/actuator networks (WSAN) that help to provide services to users. Data that comes from the nodes is integrated globally by establishing an internet connection with the WSAN's [25].

The vision of a future cyber-physical city is one that is composed of several subsystems as health, transportation, security, and others. Also, it is expected that CPS will become ubiquitous and be applied virtually to all fields as sciences, environment, agriculture and others with the help of the WASN. A WASN is an innovative sensor networks that promises to work closely with actuators to support cyber-physical systems [25].

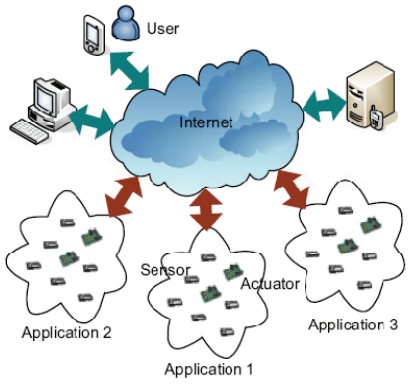

Figure 15. Physical Topology
Source: Network QoS Management in Cyber-Physical Systems

CHAPTER III

METHODOLOGY

3.1 Overview

One goal of the study is to design an architecture that allows finding a way to design a social cyber physical application for Android platform. The challenging issues are :

1) How to design an application that allows a quick response and overcome problems in latency of response, limitation of energy and others

2) How to find a way to implement location services and sensors to implement a practical application

3) Keep the application simple in order to be accepted by users.

4) How to integrate all the frameworks.

This research uses some of the most popular open sources tools to avoid licensing issues, and be able to customize the application. The main programming language will be Java for Android applications, php for the elgg-social networking server and others.

This chapter presents an architecture to follow in order to design Looc; some of the sections present justification in tools choice.

3.2 Hardware Selection

The applications are designed for T-mobile G1presented in Figure 16 or any Android phone with similar features. It is a requirement that the phone must have a GPS to help the application to obtain user location.

Figure 16. T-Mobile G1 phone

3.3 Mobile Android phone and GPS sensor

The Android SDK provides an API to receive information about available providers to obtain the current location and coordinates of the physical locations. The application requires the use of location based services (LBS) to obtain the latitude and longitude and map-based activities using Google Maps to draw the positions in the physical world. Android has a Geocoder that supports reverse and forward geocoding that are useful to convert latitude/longitude values in real-world addresses.

The device supports location services and provides its features by using the android.location package. The system service LocationManager gives an API to determine the location and use of

the capabilities. Also, the use of com.google.android.maps package is necessary to display the maps, and display options and controls for the application.

3.4 Android phones advantages and disadvantages

To design Looc the advantages and disadvantages were taken in consideration. A developer needs to understand the limitations in order to write the applications and apply the solutions around them to maximize the advantages.

Advantages

- Open source. It is not necessary to buy any license to start developing.
- Customization of the platform.
- Application with the same priority
- An Android market where a developer can offer the application to the public.

Disadvantages

- Limited energy power. When programming any Android application, saving energy is a goal, Looc is not the exception.
- Network bandwidth reliability

Android phone for its features and its openness to customization has been chosen to be the mobile device to be used to develop the application. Also, it provides the sensors to help us communicate with the physical world to generate a response.

3.5 Software selection

- Elgg. It is an open social networking platform. It can be customized according to the needs of the developer. It is chosen because it is easy to implement and administrate. Also, the application can be developed without being concerned to the limitations imposed of the other social networks.
- Android SDK integrated with Eclipse.
- Apache, MySql and php will be the tools to help and support our architecture to deploy the web and data services.

3.6 Architecture

The Looc architecture consist of the following components: android mobile client using the sensors, web services oriented to social network communication, social networking engine, and the storage engine. Those components interact with each other to be able to provide the social services. The mobile client interacts with its hardware to retrieve information from the physical world.

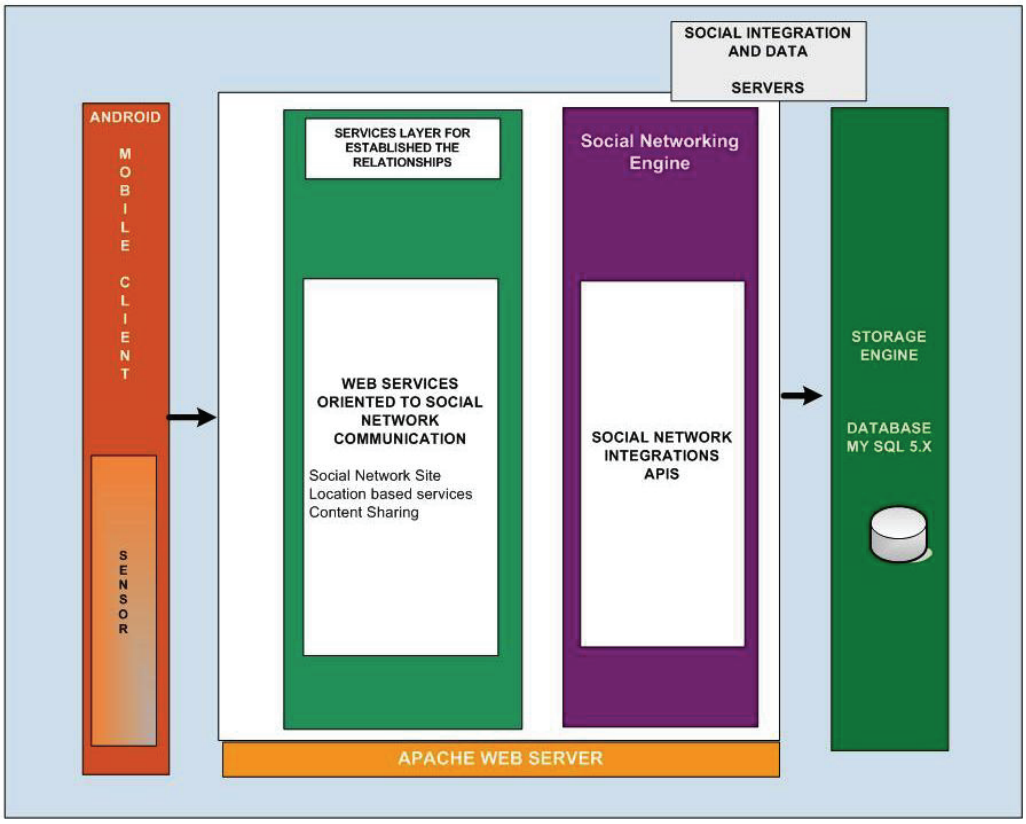

Figure 17. Looc basic Architecture

As the above figure shows the mobile client interacts with the web services to access the social network engine to retrieve the data from the storage engine. Each component is essential in the framework to retrieve the data. Also, the mobile application is a web application with native functionality because it uses native API's as geolocation. Also, the Bluetooth can provide communication with other sensors if needed.

The mobile client is a mobile social network application written for Android phone using the Android 2.2 platform. This application also interacts with the phone sensors. The client is deployed in the user mobile device. One of the main features is to connect the sensors to identify the location and send it to the social network engine.

The mobile services and the social networking engine are supported by Apache web server which provides the resources to establish the information to be read or modified. The social networking engine is elgg 1.7.4 which it helps to simulate a real social network in order to simulate the different scenarios. Also, Elgg installation offers an environment where we can work without the restrictions that other social networks in production would impose to our application. The Elgg web service provides a way to access to the social networking services as access to our contacts, sharing media and others.

The storage engine uses MySql 5.1.36. Additional functionality was added to the elgg database to access and stored the data in order to reach the goals of the application.

3.6 Potential applications

The Looc Architecture design provides a framework to access the features that combine both mobile social applications and physical world. It is designed with the intention to experiment with the geolocation services and social network applications. By simulating this environment, the goal is find new services to the users and facilitate the exchange of the communication and information. Social networking has ways to interact with our contacts and friends. However, there have been few ideas that have been experimented with people that have no relation.

The Looc architecture experiments with cyberphysical applications that can incorporate the physical world and social networks to provide new services. The goal is to find useful new experiences to potential users. To achieve this goal is necessary to rely on the mobile sensors as GPS and others. In this type of applications the interaction with the users is critical. As a future work, it is necessary to find a way to reward user's time to keep their interest. For instance, find a model that can reward the users. A solution might be to earn points redeemable for cash, make

sweepstakes and publish the results to ensure the thrush and interest of the users. Also, alliances can be made with the companies in case that the user wants to use the points on their products.

The Looc architecture has emphasis in open source technologies. This was used as guidance and a way to develop the application guarantying the maximum customization possible.

CHAPTER IV

CASE OF STUDY AND IMPLEMENTATION

4.1 Motivation

The purpose of this study seeks to research and describe the use of Android mobile applications oriented to social networks using the sensors. Also, this study explores topics related to Android mobile device, and online social networks by developing an application using the Android Software Development Kit (SDK). Currently, there have been some efforts for social networks to adapt applications using sensor in mobile phones as GPS. For instance, Twitter has decided to incorporate location feature allowing the user to display its location in the profile. One of the goals of this study is find a way to integrate location services with social networks that will help to create or explore useful applications.

It is necessary to explore new ideas in the integration of geolocation. Also, it is important to find an architecture that will help to integrate geolocation with social networks. In to complete the research the following was completed:

1. Develop a mobile android application that uses the GPS sensor to send the information.
2. Develop a plug-in to integrate the elgg social network framework to help publish the REST web Service to adjust the Elgg architecture with Looc architecture.
3. Find a way to integrate the use of the Google maps to display the geolocations.
4. Two basic cases demonstrate the potential use of the application.

4.2 Case of study

The idea in the basic scenario is first to find the location of the user A. Then find potential advisors. An advisor is a person located in a place where user A wants to know what is happening in that point.

The advisor answers the questions using also the mobile application. There are several potential uses for this type of services. For instance, a person wants to know if a parking lot is full, or what is happening in certain area. Also, a company or institution could develop its own social networking and integrate those types of services to increase the communication between their employees or volunteers. For instance, there are some institutions that send their scientists to explore an area, and sometimes they need information or pictures about a location. It would be very useful to provide services to locate potential advisor to exchange information.

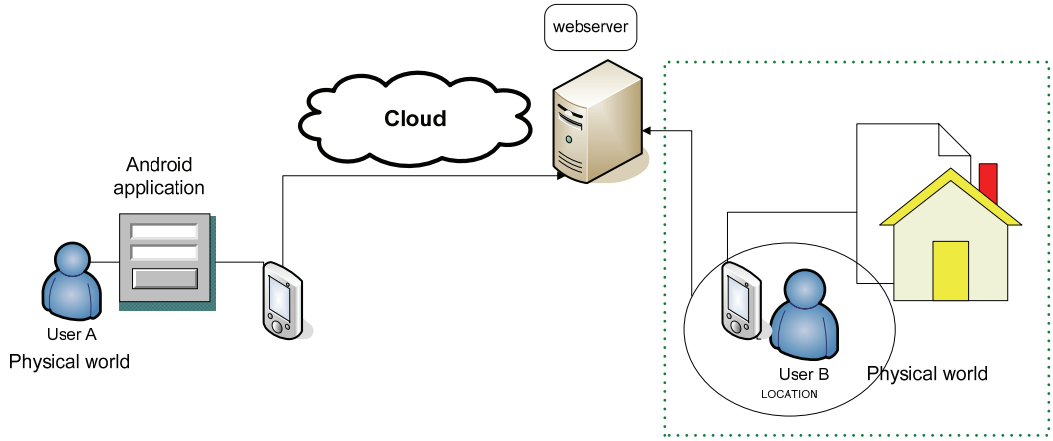

Figure 18. Basic scenario of the use Looc use

Another scenario is user A needs to know what it is happening in the surroundings. The user signs into the android application and sends its location to the server. Then, the server will submit the query according to the location specified. The query will retrieve the potential advisors to answer the question according to the coordinates provided by user A. User A now finds a potential advisor. Looc architecture provides a way that the communication between the users can be exchanged similar to a posting. The application uses Google map activities to interact with user A.

Another potential use of this type of applications is finding potential advisors in an area where a disaster has occurred. Then the advisor sends either a video or pictures. This research focuses in the basic scenario application to demonstrate that use.

4.3 Server side Implementation

The implementation consists of services oriented to social networking, database and web service connection.

4.3.1 Elgg framework

A plug-in was developed in order to add more functionality to Elgg. It is located in the subdirectory /mod. More information about how Elgg model work can be found in the website. This work is limited to discussing about the most important features from the plug-in

4.3.2 Plug-in

The plug-in is located in a subdirectory of the **/mod** folder. The name of plug-in is Looc. The start.php is the control where all the components for the plug-in are initialized. The project has web services to expose functionality in order facilitate the integration with the Android application. The plug-in contains functions that help establish relationship, retrieval of users, and other utilities.

There are three different types of web services implemented in elgg framework: users, utils, and relationship

Users: this handles functionalities as login, update location, get near by user, and get the friends, the last post of the user. Figure 19 shows the way that the login is handling for the web service. The user to log in Looc needs to be authenticated, and send the longitude and latitude. Otherwise, the user is not able to log in.

```
function looc_login($username, $password,$longitude='',$latitude=''){

$returnToken=auth_gettoken($username,$password);

if($returnToken){
$usuario=get_user_by_username($username);

$usuario->logged=true;
$usuario->save();
$returnString='<token>'.$returnToken.'</token>';
$returnString.='<guid>'.$usuario->guid.'</guid>';

return $returnString.;

}
return 'false';
}
```

Figure 19. looc_login

Utils: handles functions to verify if the user is online and the invitations to answer questions. Figure 20 shows how the function shows when the user is logged in.

```
function isUserOnline($guid){
    if($guid>0){
    $user=get_entity($guid);
    if(isset($user->logged))
        return 'true';
    }

    return 'false';

}
```

Figure 20. Is user online

Relationship: this is a very important service. It handles the advice relations. Some of the functions implemented in here are: ask advice, answer advice, get advice, get asked advice, get friend, add friend relationship, and search for looc users. Figure 21 shows the code that it is used to ask advice.

```
function ask_advice($description,$asks_strings){
    $user=$_SESSION['user'];
    $advice=new Advice;
    $advice->description=$description;
    $advice->access_id = 2;
    $saved_guid=$advice->save();

    $asks_to=split(",",$asks_strings);
    $advisors=array();
    if(count($asks_to)>0)
        foreach($asks_to as $ask_to){
            //join_group($saved_guid, $ask_to);
            add_entity_relationship($saved_guid, 'invited', $ask_to);
        }
    else{
    $ask_to=intval($asks_strings);
        if($ask_to>0){
            //join_group($saved_guid, $ask_to);
            add_entity_relationship($saved_guid, 'invited', $ask_to);
        }
    }

    if($saved_guid>0)
        return 'true';
    else
        return 'false';
}
```

Figure 21. Function asking advice

Haversine formula to calculate the distance

There are many ways to calculate the distance between two points. One is the Haversine formula which it is widely used to calculate great-circle distances between two points by using two latitude and longitude points. This method is considerable reliable [32]. The formula is presented in Figure 22.

$$d = 2\arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos_1 \cos_2 \sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right)$$

Figure 22. Harvesine Formula Source: Lewis, 2007

The following formula represents the adaptation of the formula to be applied in the query, db stands for values stored in the database.

$$d = ar\cos(\cos(\theta_1)\cos(\theta_2) + \cos(\Delta\lambda)\sin(\lambda_1)\sin(\lambda_2))$$

$$\Delta\lambda = \lambda_2 - \lambda_1$$

θ_1 = reference point

θ_2 = db

λ_1 = reference point

λ_2 = db

Figure 23. Adaptation of the Harversine formula

After finding the distance is necessary multiply by the radius of the earth which is R= 3959 mi = 6337 km. The query to retrieve the users that are around from us in the users is the following:

```
$query="select t1.owner_guid, t2.latitude, t1.longitude , ( 3959 * acos( cos(
        radians($latitude/1E6) ) * cos( radians( latitude ) ) * cos( radians( longitude ) -
        radians($longitude/1E6) ) + sin( radians($latitude/1E6) ) * sin( radians( latitude ) ) ) ) as
        distance FROM elgg_user_longitude t1 inner join elgg_user_latitude t2 on
        t1.owner_guid=t2.owner_guid HAVING distance>0 and distance<25 ORDER BY
        distance LIMIT 0 , 20";
```

Figure 24 shows the retrieval of the query. The green point represents the location of the person using Looc, the red points are the advisor who are not necessarily friends.

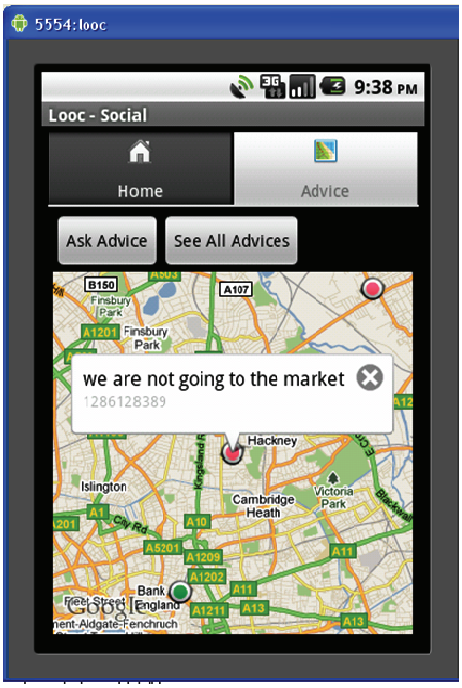

Figure 24. Query retrieval of two points

4.3.3 Database

Two views or virtual tables were created in the elgg database Figure 25. The reasons to create the views instead of changing the physical scheme of elgg database were the following:

- The query is simplified and the view works as aggregated table
- The creation of the views limits the degree of the exposure of the data increasing security.
- Views give the flexibility to change the site and the android application if the requirements change in the future.

```
mysql> describe elgg_user_longitude;
+------------+----------------------+------+-----+---------+-------+
| Field      | Type                 | Null | Key | Default | Extra |
+------------+----------------------+------+-----+---------+-------+
| string     | text                 | NO   |     | NULL    |       |
| owner_guid | bigint(20) unsigned  | NO   |     | NULL    |       |
| longitude  | double               | YES  |     | NULL    |       |
+------------+----------------------+------+-----+---------+-------+
3 rows in set (0.00 sec)

mysql> describe elgg_user_latitude;
+------------+----------------------+------+-----+---------+-------+
| Field      | Type                 | Null | Key | Default | Extra |
+------------+----------------------+------+-----+---------+-------+
| string     | text                 | NO   |     | NULL    |       |
| owner_guid | bigint(20) unsigned  | NO   |     | NULL    |       |
| latitude   | double               | YES  |     | NULL    |       |
+------------+----------------------+------+-----+---------+-------+
```

Figure 25. Table Views of Elgg Database

4.4 Android application

The android application handles the authentication initiation with the server, friend's retrieval, advice, use the map, retrieving and sending the coordinates to elgg database.

Authentication: Figure 26 shows the login function. The android application calls the request method to make a request through Android and retrieve the answer. When the user gets authenticated, the application sends the location to be stored it in the elgg database Figure 27.

```
private void login() {

    // we get the user text
    String login = textUser.getText().toString();
    String password = textPass.getText().toString();
    if (!login.equals("") && !password.equals("")) {
        // create a user entity
        User user = AuthenticationService.authenticate(LoocLogin.this
                .getBaseContext(), login, password);
        /* Create the message to say to the UI get out of here */

        Message message = new Message();
        if (user.isAutheticated()) {
            message.what = LoocLogin.SUCCESS;

        } else {
            message.what = LoocLogin.UNSUCCESS;
        }
        messageHandler.sendMessage(message);
    } else {
        // TODO: The login or password are empty string no good for us
    }
}
```

Figure 26. Login to Looc

```
    LocationManager locationManager = (LocationManager) context.getSystemService(Context.LOCATION_SERVICE);
    Location location = locationManager.getLastKnownLocation(LocationManager.GPS_PROVIDER);

    if(location!=null ){

        double longitude=location.getLongitude()*1E6;
        double latitude=location.getLatitude()*1E6;

        Log.i("Longitude/Latitude",location.getLongitude()+"/"+location.getLatitude());
        Log.i("Longitude/Latitude",String.valueOf(longitude)+"/"+String.valueOf(latitude));
        datas.clear();
        datas.add(new BasicNameValuePair("auth_token", userHandler.getUser().getAuthToken()));
        datas.add(new BasicNameValuePair("longitude", String.valueOf(longitude)));
        datas.add(new BasicNameValuePair("latitude", String.valueOf(latitude)));
        xmlString=requester.request("users.updateLocation", datas);
    //Log.i("Answer",xmlString);

    }


    setCurrentUser(userHandler.getUser());

    return user;
}
```

Figure 27. Location update

Friends: Figure 28 shows a piece of the code in the application. The function here is to retrieve a list of users that has a relationship as friends. It uses the web service located in users.php in the elgg plug-in and authentication is required. The method returns the list of friends of the user authenticated in the Android application.

```
public List<Friend> getMyFriends(String guid) {
    // we get the singleton instance of user
    User user = AuthenticationService.getCurrentUser();

    List<NameValuePair> datas = new ArrayList<NameValuePair>();
    datas.add(new BasicNameValuePair("auth_token", user.getAuthToken()));

    if(guid!=null){
        datas.add(new BasicNameValuePair("guid", guid));
    }

    InputStream xmlInput = requester.requestInputStream("users.friends",
            datas);



    return FriendParser.getFriends(xmlInput);

}
```

Figure 28. Method get Friends

Map: Figure 29 shows how to obtain the potential adviser to be displayed later in the map activity. Also the class Friend in the android application has a Boolean to help us to distinguish friends from users with no relation.

```
// first overlay
drawable = getResources().getDrawable(R.drawable.marker);
itemizedOverlay = new LoocOverlay(this,drawable, mapView);

List<Friend> nearByFriends = new FriendService(this).getNearByUser(0,
        null, null);
for (Friend nearByFriend : nearByFriends) {
    Double latitude = Double.parseDouble(nearByFriend.getCoordinate()
            .getLatitude()) * 1E6;
    Double longitude = Double.parseDouble(nearByFriend.getCoordinate()
            .getLongitude()) * 1E6;
    GeoPoint point = new GeoPoint(latitude.intValue(),
            longitude.intValue());
    Post post = nearByFriend.getLastPost();
    OverlayItem overlayItem = null;
    if (post != null)
        overlayItem = new LoocOverlayItem(point, post);
    else
        overlayItem = new OverlayItem(point, "", "");
    itemizedOverlay.addOverlay(overlayItem);
    Log.i("L/L", " - " + latitude + "/" + longitude);
}

mapOverlays.add(itemizedOverlay);
```

Figure 29. Obtain the potential adviser and locate them in Map

Advisor: advisor is located in a ratio less than 25 miles of the point the user is interesting to ask a question. To send the message, the application use advice service to send it to the database. The 25 miles was chosen just for purpose of testing. It can be set to a value less than 25.

```
private void askAdvice() {

    txtAdvice = (TextView) findViewById(R.id.editAsk);
    String[] asksString=null;
    if(asksTo!=null){
        asksString=new String[asksTo.size()] ;
        for(int i=0;i<asksTo.size();i++){
            asksString[i]=asksTo.get(i).getGuid();
        }

    // TODO: place the advice friend UI
    boolean sendAdvice = adviceService.sendAdvice(txtAdvice.getText()
            .toString(), asksString);
    Message message = new Message();
    if (sendAdvice) {
        message.what = AskAdvice.SUCCESS;
    } else {
        message.what = AskAdvice.UNSUCCESS;
    }
    messageHandler.sendMessage(message);
    }
}
```

Figure 30. Ask advice method

4.5 Validation Results and Analysis

Figure 31 shows the normal execution when the user log in the android application and look for advice to send a question. The user needs to be authenticated in order to start using the application. As soon as the user is authenticated the location is sending it to the server to be stored. For this flowchart, the user log-in, selects an advisor from the map and send a question Figure **32** shows the map, green point represents the user geographic point and the red points represent the potential advisor. A notification is sent it to the potential advisor to answer the question. When the advisor completes the answer, the application sends it to the elgg server. Then, it sends a notification to the user that he/she has its answers.

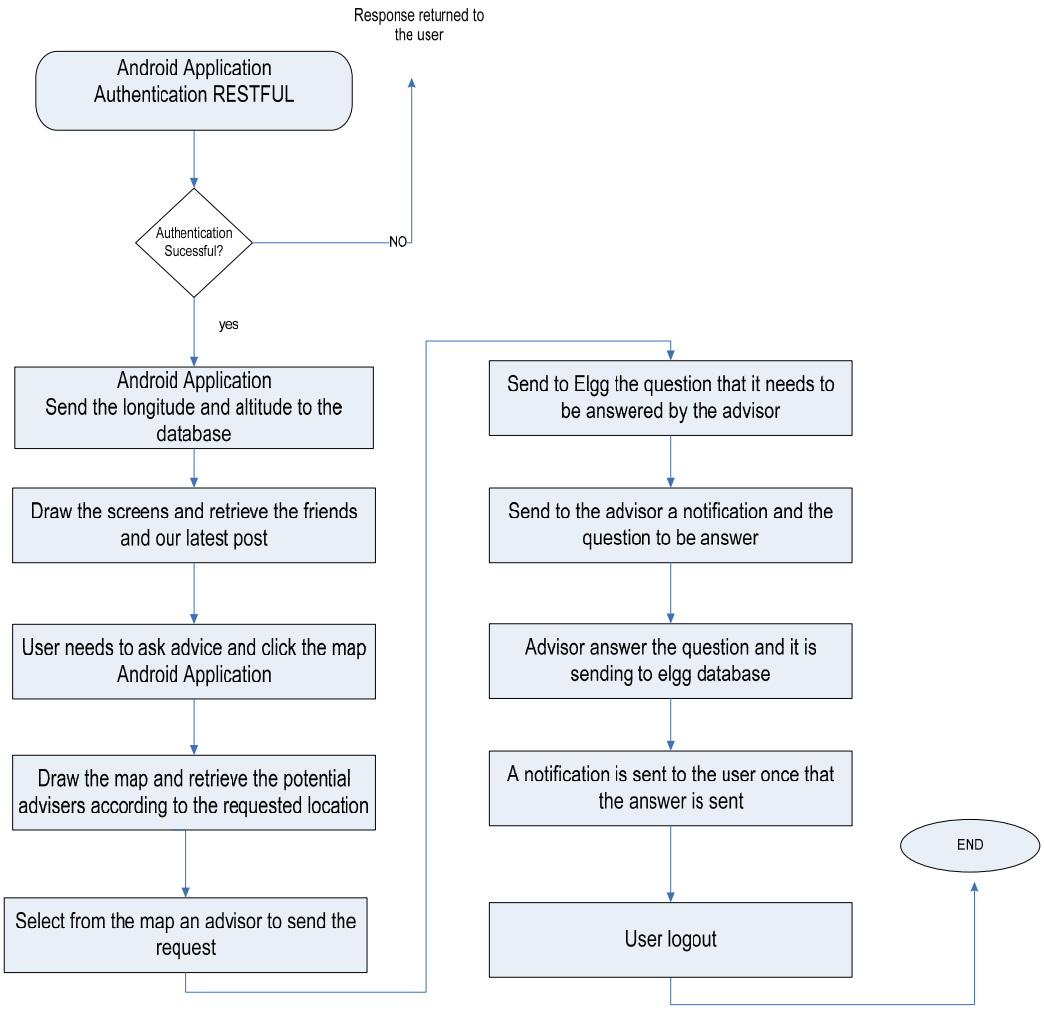

Figure 31. Android use of the advice feature

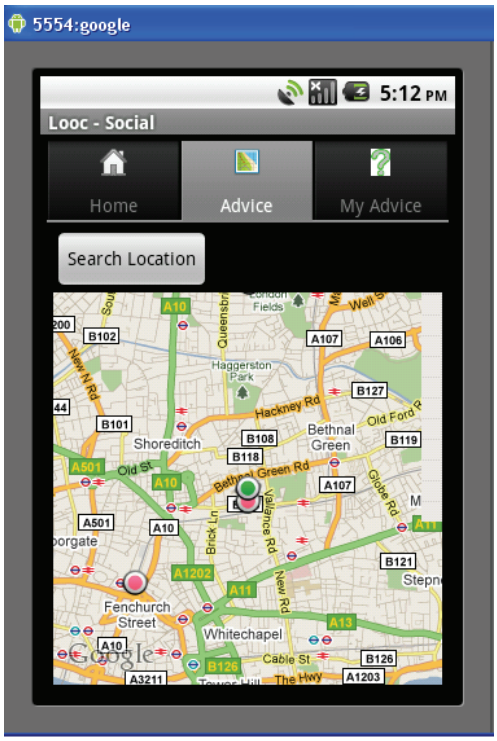

Figure 32. Advisor maps according to the location

Figure 32 shows the map view. The green point represents the location of the user A, the red points represents other users who can be potential advisors. The first action to take place when you click on the map view is to find the user location and find potential advisors. This is our first case when the user wants to know what it is happening around.

Our second case happens when a user wants to know what it is happening using a specific address. In Figure 32 there is a button in the map view, Search Location button where the user can type the address in order to find potential advisors in that area. Figure 33 shows the dialog where user can type the address. The map view will be similar to the figure 32 and it will display the potential advisors in that area.

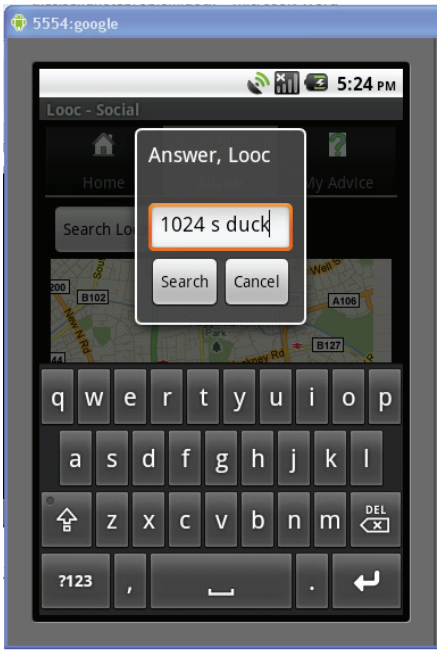

Figure 33. Search location for typing address

Figure 34 shows the potential advisor (red point) for an area for the basic scenarios. To ask for an advice the user needs to double click in the area to retrieve the potential advisors according to the point clicked. A red balloon represents that the user A wants to ask a question according to that location. Yellow balloons represents advice already answered in that zone.

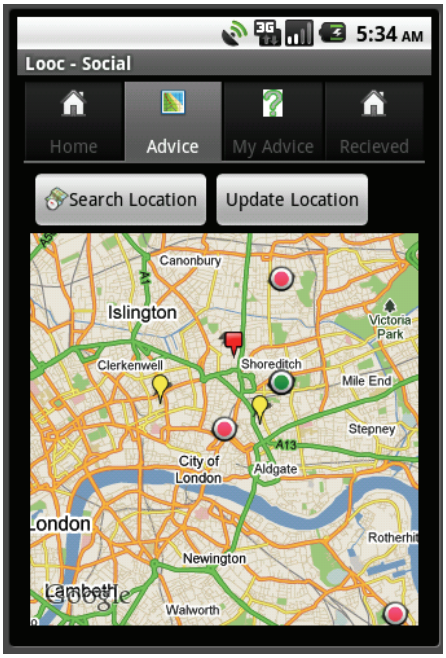

Figure 34. Potential advisor in the area

Then user A clicks in the potential advisor by clicking over the red point. The user is displayed, for the purposes of this research and demonstration, the picture of the user is displayed. However, a good privacy analysis must be done in the future to preserve the privacy of the user. In the application, there is an option where an avatar can be displayed instead of the user profile picture. User A can type the question and click on ask to send it to the potential advisor. Then, Figure 36 shows user A screen with an answer of the advice asked.

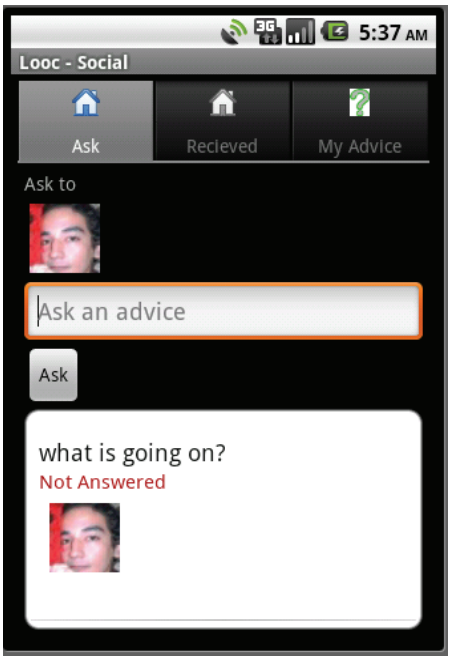

Figure 35. Potential advisor

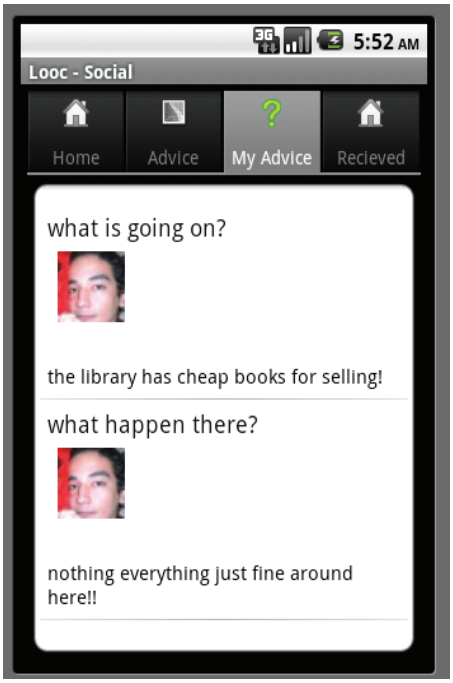

Figure 36. User A Screen

Finally, the integration and interaction with all the components can be summarized in Figure 37. Android application uses the services of the cloud to access the web services deployed in Elgg. Elgg sends the response by sending a file in xml that the android application must parse to obtain the answer sent by the elgg.

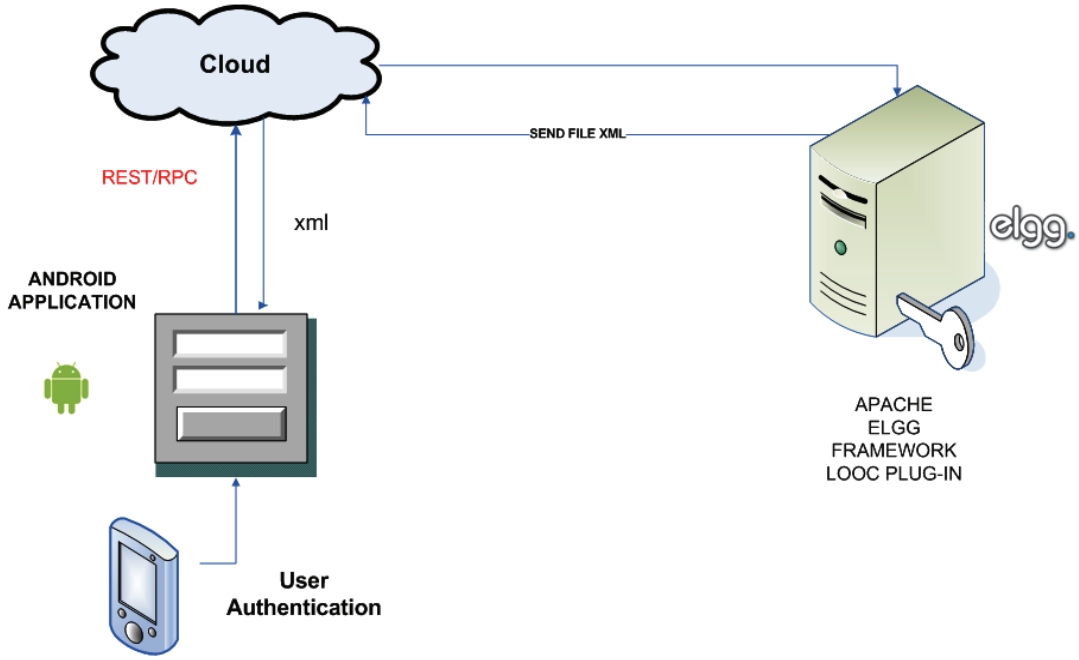

Figure 37. Integration with all the components

CHAPTER V

CONCLUSIONS AND FUTURE WORK

This study shows the challenges related to develop mobile cyber-physical social network with android. The prototype of Looc was developed and integrated into other frameworks to demonstrate its usage. One of the most important challenges was find a good communication method for the social network and an appropriate way to locate individuals. Haversine formula was used to calculate the distance in our prototype with the purpose to demonstrate its use.

One of the challenges for social cyber-physical android applications has to face is the best use of the energy and resources. Several techniques were used to help save the battery and other resources in the demo. Android phones notification system is useful to let the user know that the application requires attention. Other challenge to overcome for cyberphysical social network application is the integration of different frameworks and programming languages. A developer in cyber-physical systems needs to be aware of several techniques and resources and find ways to apply them.

Also, for cyber-physical social system the key to be successful is to answer the question about how to make a user to contribute and exchange experiences. One approach could be offering goods or services. It would be interesting for future work to explore how to merge this project in a way that can be attractive to use it. Especially now when there are more people acquiring smart phones with GPS features and other type of sensors. However, privacy and legal issues must be addressed to help to integrate this type of features. It is necessary to explore how to protect user's identity even if the user is a friend. For this research, we protect the user and advisor identity by not supplying sensitive information that can reveal the person gender, description, and name just an avatar is displayed in the demo. Those boundaries and security issues are worth to further investigation.

REFERENCES

[1] I. T. I. Statistics. (2008, 2009-06-09 17:17:36). *Worldwide mobile cellular subscribers to reach 4 billion mark late 2008*. Available: http://www.itu.int/ITU-D/ict/newslog/Worldwide+Mobile+Cellular+Subscribers+To+Reach+4+Billion+Mark+Late+2008.aspx

[2] P. Leo. (2006, 06-09-2009). *Cell phone statistics that may surprise you*. Available: http://www.post-gazette.com/pg/06075/671034-294.stm

[3] "Open Handset Alliance," ed, 2008.

[4] C. Yao-Jen*, et al.*, "A General Architecture of Mobile Social Network Services," in *Convergence Information Technology, 2007. International Conference on*, 2007, pp. 151-156.

[5] J. F. DiMarzio, *Android : a programmer's guide*. New York: McGraw-Hill, 2008.

[6] R. Meier, "Professional Android application development," ed. Indianapolis: IN : Wiley, 2009, p. 409.

[7] (2008). *Android Developers*. Available: http://developer.android.com/index.html

[8] C. N. Baker, Danny. (2008, Using Android in Education for Mobile Device Development. Available: http://www.eng.auburn.edu/users/bakercr/Final_Paper.pdf

[9] F. W. Ableson*, et al.*, *Unlocking Android : a developer's guide*. Greenwich: CT : Manning, 2009.

[10] R. Rogers, *Android application development*. Cambridge: O'Reilly, 2009.

[11] (2008, 2009-07-03 20:17:43). *Android Debug Bridge*. Available: http://developer.android.com/guide/developing/tools/adb.html

[12] B. Speckmann, "The Android mobile platform," Master Computer Science, Eastern Michigan University, Ypsilanti, 2008.

[13] F. Y. Wang*, et al.*, "Social computing: From social informatics to social intelligence," *Ieee Intelligent Systems,* vol. 22, pp. 79-83, Mar-Apr 2007.

[14] A. Siemer and S. M. Swafford, "ASP.NET 3.5 social networking : an expert guide to building enterprise-ready social networking and community applications with ASP.NET 3.5 /," ed. Birmingham, UK: Birmingham, UK, 2008, p. 556.

[15] H. B. Hu and X. F. Wang, "Evolution of a large online social network," *Physics Letters A,* vol. 373, pp. 1105-1110, Mar 2009.

[16] danah m. boyd and Nicole B. Ellison, "Social Network Sites: Definition, History, and Scholarship," *Journal of Computer-Mediated Communication,* vol. 13, pp. 210-230, 2008.

[17] A. Acquisti*, et al.*, "Imagined communities: awareness, information sharing, and privacy on the Facebook," in *Privacy Enhancing Technologies. 6th International Workshop, PET 2006. Revised Selected Papers (Lecture Notes in Computer Science Vol.4258)*, Place of Publication: Berlin, Germany; Cambridge, UK. Country of Publication: Germany.

[18] C. Pascu, "An Empirical Analysis of the Creation, Use and Adoption of Social Computing Applications. IPTS Exploratory Research on the Socio-economic Impact of Social Computing," European Commission Joint Research Centre Institute for Prospective Technological Studies EUR 23415, 2008.

[19] E. Karjaluoto, "A primer in Social Media examining the phenomenon, its relevance, promise and risks," 2008.
[20] E. C. Ortiz. (2008, 2009-07-18). The Mobile Context and People-Centric Mobile Computing An Introduction to the mobile context and mobile social software. Available: http://cenriqueortiz.com/pubs/themobilecontext/TheMobileContext-CEnriqueOrtiz.pdf
[21] C. Honey and S. C. Herring, "Beyond Microblogging: Conversation and Collaboration via Twitter," in *System Sciences, 2009. HICSS '09. 42nd Hawaii International Conference on*, 2009, pp. 1-10.
[22] O. Garden. (2005, 07-19-2009). *Mobile video podcasting: The killer app for mobile video?* [Electronic]. Available: http://opengardensblog.futuretext.com/archives/2005/07/mobile_video_po.html
[23] R. K. Ganti*, et al.*, "Sense world: Towards cyber-physical social networks," St. Louis, MO, United states, 2008, pp. 563-564.
[24] G. Karsai and J. Sztipanovits, "Model-integrated development of cyber-physical systems," Anacarpi, Capri Island, Italy, 2008, pp. 46-54.
[25] F. Xia*, et al.*, "Network QoS management in cyber-physical systems," Chengdu, Sichuan, China, 2008, pp. 302-307.
[26] "NSF Workshop on Cyber-Physical Systems."
[27] K.-D. Kang and S. H. Son, "Real-time data services for cyber physical systems," Beijing, China, 2008, pp. 483-488.
[28] "The Distributed Robotics Garden."
[29] E. A. Lee, "Computing needs time," *Communications of the ACM,* vol. 52, pp. 70-79, 2009.
[30] E. A. Lee, "Cyber physical systems: Design challenges," Orlando, FL, United states, 2008, pp. 363-369.
[31] B. D. Noble and J. Flinn, "Wireless, self-organizing Cyber-Physical Systems," 2009.
[32] A. Lewis, *Beginning Google maps applications with Rails and Ajax : from novice to professional*. Berkeley : Apress: New York, 2007.

VITA

Carmen Patricia Ayerdis Espinoza

Candidate for the Degree of

Master of Science

Thesis : LOOC: A CYBER-PHYSICAL SOCIAL NETWORK ON ANDROID PLATFORMS

Major Field: Computer Science

Biographical:

Personal Data: Born in Managua, Nicaragua, Female

Education: Completed the requirements for the Master of Science in Computer Science at Oklahoma State University, Stillwater, Oklahoma in December, 2010. Completed the requirements for the Bachelor of Science in computer System Engineering in UNICA Universidad Catolica in Managua, Nicaragua in 2000.

Experience: Prior to come OSU, worked for the Nicaragua Government National Agency for the protection of the Natural Resources and the Environment (MARENA) as a Help Desk Support Manager

Name: Carmen Patricia Ayerdis Espinoza Date of Degree: December, 2010

Institution: Oklahoma State University Location: Stillwater, Oklahoma

Title of Study: LOOC: A CYBER-PHYSICAL SOCIAL NETWORK ON ANDROID PLATFORMS

Pages in Study: 72 Candidate for the Degree of Master of Science

Major Field: Computer Science

Scope and Method of Study: Thanks to the increasing use of mobile devices, the mobile phone applications area is quickly developing as the popularity and demand for new features. This study has the goals to explore the integration between the android platform using sensors with social networks. Architecture to develop this type of application is also proposed. Within the document a literature review is provided and is composed of four parts: the first part includes the necessary concepts to program Android applications with best practices, taking in consideration its life cycle, the second part discusses the mobile social computing concepts and challenges, and the last part shows the main concepts and challenges in cyber-physical systems. Thereafter, a methodology and architecture is proposed in chapter 3. The chapter 4 includes two case scenarios and its implementation which has android application and social network.

Findings and Conclusions: The study shows the challenges related to develop mobile cyber-physical social network with android. A demo in Android was developed and integrated it to other frameworks to demonstrate its use. One of the most important challenges was find a good communication method for the social network and an appropriate way to locate individuals. Haversine formula was used to calculate the distance in our demo. However, it is necessary to explore more methods that are more exact in calculate the distance between geographical points. Also, privacy and legal issues are necessary to be addressed for mobile cyber-physical social networks in the future.

ADVISER'S APPROVAL: Xiaolin Li