3

Mobile Application GPS-Based

Berta Buttarazzi
University of "Tor Vergata", Rome, Italy

1. Introduction

Most of navigators for mobile devices have a big failure; they do not notify the user of road condition (traffic, checkpoints, car accidents, road work) unless the user has some kind of paid subscription. Even in this case the only information that are provided are about traffic. Many famous Navigators notify user of traffic using sensors under the road, but it is not a real time update. Besides there is virtually no way to notify user of car accidents or checkpoints at real time, user usually downloads POIs (Points of Interest) from Internet and uploads them to the Navigator which will display them on the map.

Road condition change very frequently so POIs should be updated faster so that an user can decide to take another road to avoid traffic, to reduce pollution, to save time. An humble application like this can also be used in a very useful way to get our life better!

The purpose of this work has been to study and develop a way that displays updated POIs on a map. Thanks to the Web 2.0 philosophy the best way that has been found is to allow users to report any events on the road. Users can also report if an event no longer exists so that markers on the map can be updated in the fastest and most accurate way possible. Because of high interaction between users, this application is like a Social Network where the relationship between users are based on GPS positions.

Mobile devices are obviously the best devices to do that, thanks to their integrated GPS sensor and to the easy development environment like Android SDK. Android has been chosen because it is completely open source and because it is a Google OS and it can be used in the best way in conjunction with Google Maps.

2. Environment

2.1 MOBILE WEB 2.0

World Wide Web was born in 1993 and in the latest 20 years there is a completely new way of using it. After Web 2.0 philosophy has developed since 1999, new technologies and new ways of approaching the Internet were born.

In Web 1.0 Users could only access website to retrieve information. These information were written by the webmaster and read by the users. Web 2.0 introduced a fast change of approach. Contents were created not only by webmasters, but also by the users. Broadband connection helped a lot users to connect to the Internet and to publish their contents.

Websites like Wikipedia, YouTube and Flickr had a large impact on World Wide Web. developments of Web 2.0.

We can say that the biggest differences between Web 1.0 and Web 2.0 are the following:

- Web 1.0 was about reading, while Web 2.0 is about writing;
- Web 1.0 was about client-server, while Web 2.0 is about peer to peer;
- Web 1.0 was about wires, while Web 2.0 is about wireless;
- Web 1.0 was about home pages, while Web 2.0 is about blogs;
- Web 1.0 was about web forms, while Web 2.0 is about web applications;
- And many others…

Users started to share their thoughts, their ideas and their multimedia content on these websites and they start to interact between each other more and more. In 1997 this interaction had a blast when Social Network like Facebook started to have a lot of users.

A Social Network is a social structure made up of individuals which are connected by one or more interdependency, like friendship in Facebook. It is important to notice how much Social Network are used nowadays.

Between the end of 2007 and the beginning of 2008 mobile devices with Apple OS and Android OS entered the market. A new kind of Web was born; everything that could be done in our home in front of a PC could be done everywhere by using a smartphone. Besides new technologies created new ways to use the Internet. GPS Sensor embedded in the smartphone could help locate the user's position so navigation via satellite could be done by simple smartphones instead of ad hoc navigator (like TomTom).

In 2011 smartphones sales beat PC sales for the first time in history. It is an important change, we are not talking about Web 2.0, but we are in the Mobile Web 2.0 era.

2.2 Android

Android is an operating system for mobile devices such as smartphones and tablet. It consists in a Kernel based on Linux Kernel, with middleware, libraries and APIs written in C and application software running on application framework which includes Java compatible libraries. Uses the Dalvik Virtual Machine to run compiled Java Code.

Current version is now the 3.2 Honeycomb released in July,2011, which is an incremental release that adds new capabilities like zoom-to-fill screen compatibility mode; capability to load media files directly from the SD card.

It uses SQLite to store datas, OpenGL ES 2.0 for 3d Graphics. It includes GPS Sensor, Accelerometer, Gyroscope, Magnetomer, Proximity and Pressure Sensor, Thermometers. Supports multi-touch and Bluetooth and is multitasking.

It provides an Eclipse Tool inside the SDK to help developer program in Eclipse Environment.

3. Application

The application is divided into 2 smaller applications:

1. Android Application: it is on the mobile device and it displays current information on Google Maps, allows user to report any events by tapping the map, updates Social Network status.
2. Server Application: it communicates with the DB where all the information about users and events are stored. Android Application and Server Application communicate between them with a SSL connection.

3.1 Android application

It is the biggest and most important part of the entire application. It uses Android SDK API to manage the GPS Sensor, Google Maps API to show the Map powered by Google Maps, to display the markers about events on the Map. The other classes must send information to the Server or to Facebook/Twitter App.

The first and most important part is acquiring the User's Location. It's important to manage it properly because the aim is to get the most accurate location and use the least battery possible.

Android utilizes GPS and Android's Network Location Provider to acquire the user location. Although GPS is most accurate, it only works outdoors, it quickly consumes battery power, and doesn't return the location as quickly as users want. Android's Network Location Provider determines user location using cell tower and Wi-Fi signals, providing location information in a way that works indoors and outdoors, responds faster, and uses less battery power.

Obtaining user location from a mobile device can contain errors and be inaccurate due to:

- Multitude of location sources: determining which to use and trust between GPS, Cell-ID, and Wi-Fi;
- User movement: location changes and we need to re-estimating user location every so often;
- Varying accuracy: a location obtained 10 seconds ago from one source might be more accurate than the newest location from another or same source.

Android provides a best performance model to obtain user location:

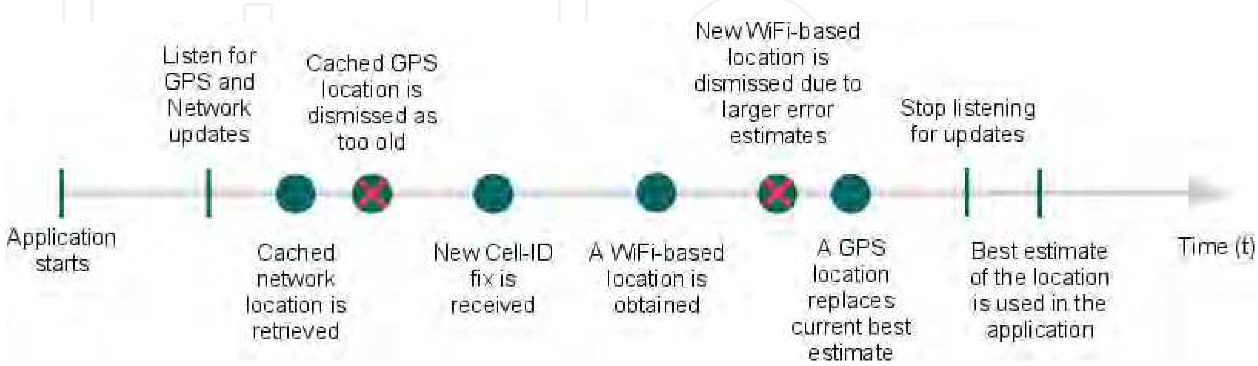

Fig. 1. Timeline representing the window in which an application listens for location updates

The application will start listen to location updates just after it started. After a certain amount of time the application will come back to listen to updates and will estimate the best location between the old one and the one just retrieved.

The best way to get the most accurate location is to use the API method *getAccuracy()*. This method returns the accuracy of the fix in meters. If for example this method returns an *Accuracy* value of 50, it means that user is in a 50 meters radius from the position that is shown on the map.

However if we want a location to be the most accurate possible, we have to use a lot of battery, so we should save some of our battery and have a less accurate location.

There are several tricks we can use to save device battery which is fundamental in this application:

1. Reduce the size of the window above in which we listen to location updates, less time the GPS sensor will be on the less battery will be used;
2. Using a last known location to increase the speed the location is shown on the map (every first time the application is launched, it takes a while to get a fix of the current location, because the GPS sensor must connect to several satellites);
3. Requesting location updates less frequently;
4. Use just one of the two providers (GPS and Network).

Obviously these tricks have a con: they don't help us to get a very accurate location. But the objective is to have the application works for a long time.

To achieve the most valuable result this application uses the following algorithm to save device's battery:

1. Check if the retrieved location is more recent than the one we are using by *getTime()*;
2. Check if the retrieved location has a better Accuracy than the one we are using by *getAccuracy();*
3. Check which provider is more dependable and use a location retrieved by that provider.

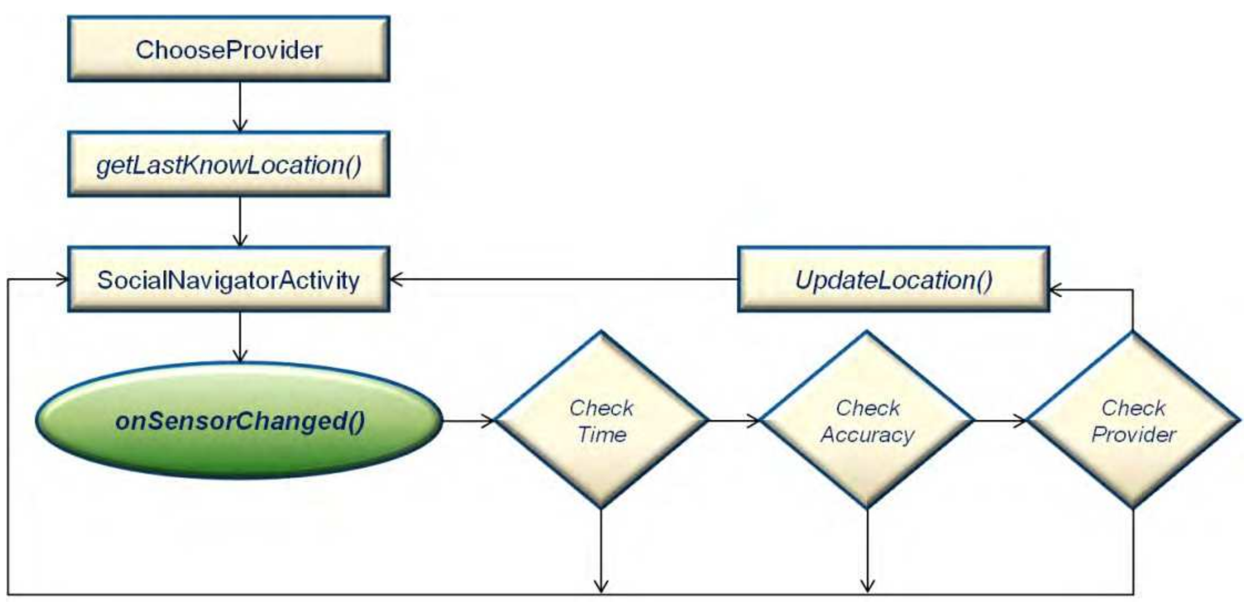

Fig. 2. Algorithm to choose the best location saving battery.

The problem is when the GPS sensor is off or when there is no GPS signal (i.e. when we are in buildings). The Android Network Provider will first get the Cell ID and then will send it to google server, which maps such Cell IDs and the server will return a latitude and longitude. In this case accuracy is usually very low, for example 1000 meters. By this time Android will also try to see all WiFi networks in the area and will send information about them too to the google server and if possible google server will return a new location with higher accuracy for an example 800 meters. It is very important to notice that the Network Provider requires an internet connection. So if the user is in a building and there is no data connection, he will never be able to get a location.

Once the application has the user's location, it will connect to the server and download the list of events and their location and it will show a map centered in the user's location and will display the markers of the events that are around him, as shown in the next figure.

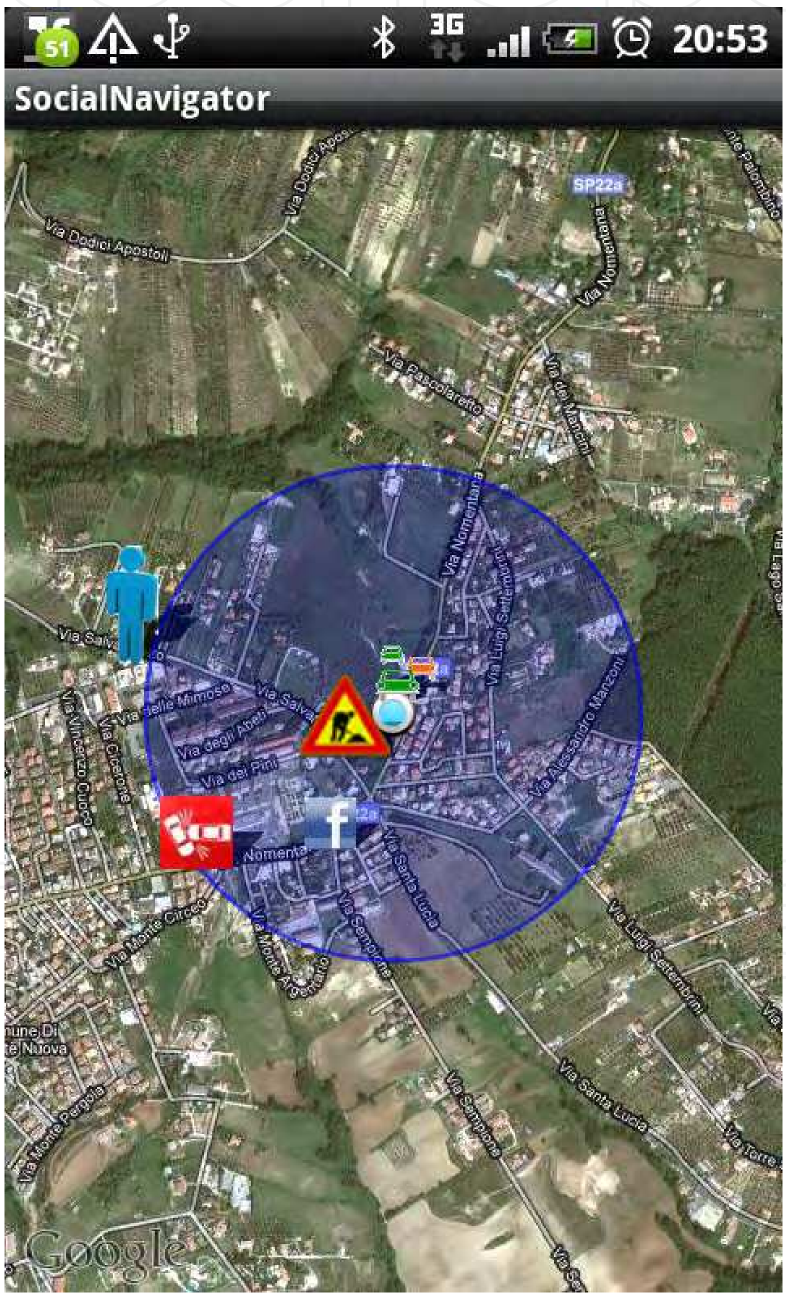

Fig. 3. View of the Map on the Mobile Device with the markers of traffic, road works, car accidents, FB events and friends

In the above figure we can see different markers:

- Our position in the center;
- One of our friends;
- A car accident;
- Traffic jam;
- Road Work;
- A Facebook event.

The user can tap on any of these events to see more information or delete them if he realizes that the reported event is no more in that location.

Fig. 4. Info Dialog about an event (in this case an accident).

Events like car accidents will be automatically deleted by the server after 24 hours after they have been reported.

If an user wants to report a new event, he can tap twice on the map and the application will ask him what kind of event he is reporting. After that the application will compose a message and send an update request to the server.

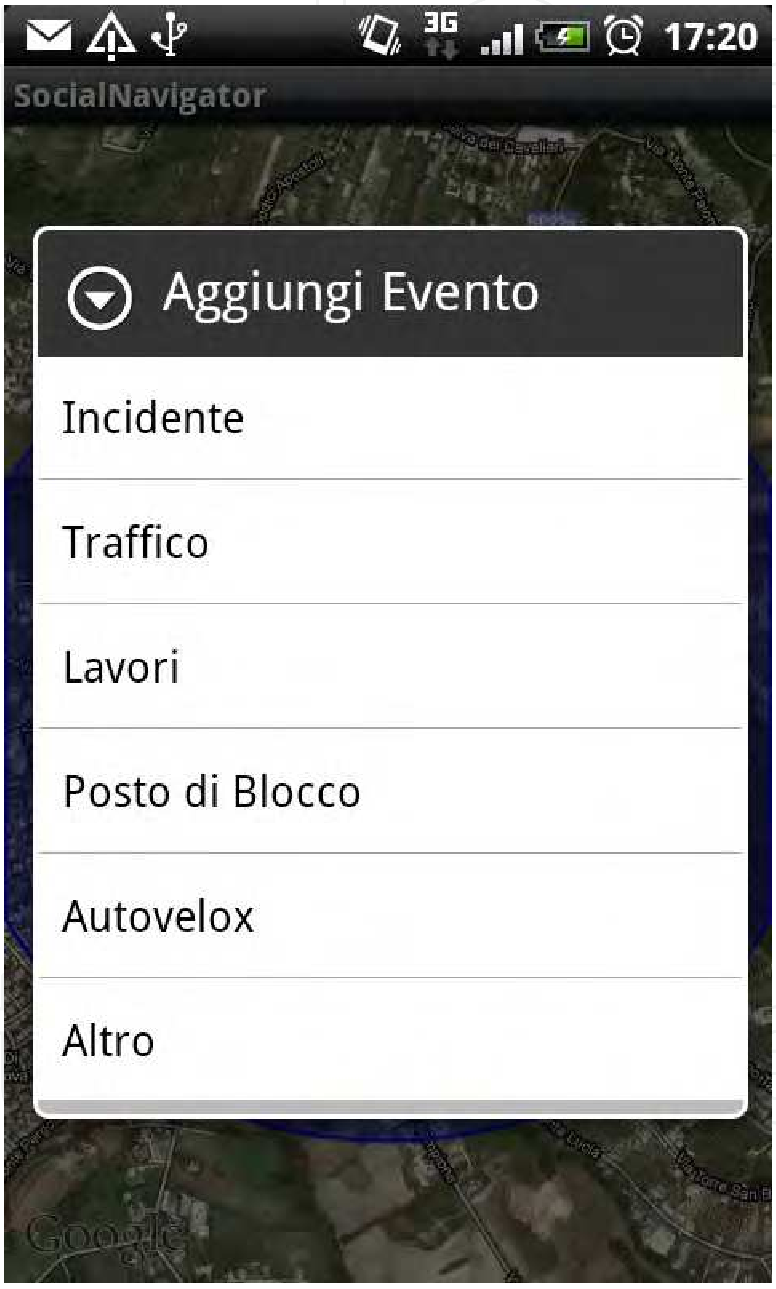

Fig. 5. Dialog to report a new event.

The application can show position of user's friends (after downloading the friend list from Facebook). The information that an user can see by tapping on a people marker are name and phone number and he can decide to call or text his friend from inside the app. The application will also notify the user if any of his friends is close to him so that he can call them.

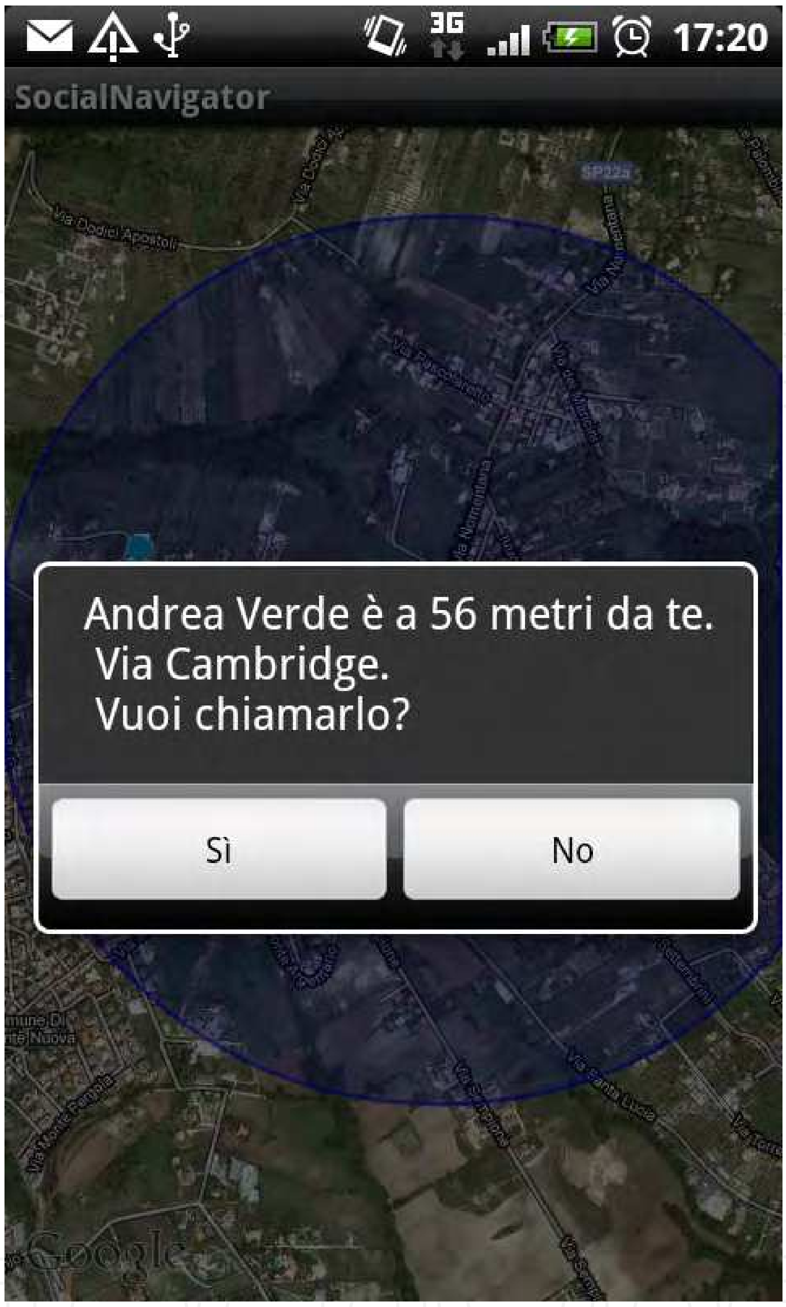

Fig. 6. Dialog that report a friend's position near to us

It is also important to give the user the chance to manage his privacy, so he can decide which markers he wants to see and he can decide if he wants his friends to see his location on their Map. Besides he can also decides if he wants to share his reports on Social Network like Facebook and Twitter.

Thanks to Android API, it is possible to check the speed an user is moving using the GPS Sensor so the application can understand by itself if there is traffic on the road. Obviously there is the chance that the user has momentarily stopped and there is not traffic, so if the speed decreases for a certain time under 15 km/h the application will ask the user if he is in a traffic jam. In case of a positive answer (user will see a dialog and will just have to click a YES or NO button) the application will update the server with the location of a traffic jam.

Fig. 7. Privacy Preferences

One of the big problem is that the original MapView doesn't allow the user to add anything on it, but it only handles zoom in/out. The idea was to handle three different types of interaction between user and screen:

1. One single tap on a marker that opens a dialog with information about the event
2. Two taps on the map to open a dialog from which user can choose what kind of event report
3. Zoom in/out in the same way of all the other smartphones, using two fingers.

Login into the applications is very simple, Android stores user's Google credentials and the application uses them to retrieve user info, like name and friends.

3.2 Server application

This application must communicate with the application on the mobile device. It will receive data about new or old (if they have to be removed) events and it will update the DB. Every type of event has its own table in the DB, and every record has latitude, longitude, date/time and id of the user that has reported it. The People Table is a little more articulate because it has also the Facebook friend list and all the permissions the user has granted.

The events that the server application must handle are:

- Mobile device notifies user's data to the server;
- Mobile device notifies its location to the server;
- Mobile device requests to retrieve events' list;
- Mobile device requests to update events' information.

4. Conclusion

This application overcomes many limitations of Navigator and Social Network. It can be seen as a point of union between a Navigator and a Social Network. It proves the use of GPS not only for user's advantages but for more people because reports about road condition can be seen also by people that don't use the application but have access to user's Facebook/Twitter profile.

It helps users to avoid traffic and to choose another road so it can be useful not only to reduce traffic but also to reduce pollution and to protect environment.

Its strength and at the same time limitation is the fact that it is a mobile application. So it can completely use the GPS but that means it will use a lot of battery. If battery duration will increase, this kind of application could be on every mobile device and a lot of other markers could be added.

In future there can be a lot of good developments, for example it can be useful use vocal commands to report events. It could also be implemented VOIP and data connection between users. The limitation of these future developments is in the lack of APIs by Android and programs for VOIP like Skype.

5. References

Book

Reto Meier, 2010, *Professional Android 2 Application Development*. Wiley Publishing Inc., Indianapolis, USA

M. Carli, 2011,*Android 3 Guida per lo sviluppatore*, Apogeo, Italy

Endarnoto, Pradipta, Nugroho, Purnama,2011, *Traffic Condition Information Extraction & Visualization from Social Media Twitter for Android Mobile Application* , International Conference on Electrical Engineering and Informatics, 17-19 July 2011, Bandung, Indonesia

Russello, Crispo, Fernandes, Zhauniarovich*, 2011, YAASE: Yet Another Android Security Extension,* IEEE SocialCom 2011 & PASSAT 2011, October 9-11, 2011, MIT, Boston, USA

D.Rosenblum, 2007, *What Anyone Can Know: The Privacy Risks of Social Networking Sites*, Security & Privacy, IEEE, May-June 2007, Volume: 5 Issue:3- pages: 40 - 49

Website

http://developer.android.com/index.html

Advanced Topics in Multimedia Research

Edited by Dr. Sagarmay Deb

ISBN 978-953-51-0078-2
Hard cover, 104 pages
Publisher InTech
Published online 17, February, 2012
Published in print edition February, 2012

As multimedia has become a very important technology, significantly improving people's lives, this book provides an up-to-date scenario of various fields of research being carried out in the area. The book covers topics including web-based co-operative learning, effective distance learning through multimedia, quality control of multimedia on the internet, recovery of damaged images, Network-on-Chip (NoC) as a global communication vehicle, and Network GPS for road conditions (such as traffic and checkpoints). We believe that the book will help researchers in the field to proceed further in their research on multimedia.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821