“Know” Your Mobile Customers: A Design Approach To An Android-Based Mobile Analytics Tool

Qi Qi Jiang
Copenhagen Business School, qi.digi@cbs.dk

Lele Kang
Nanjing University, lelekang@nju.edu.cn

Follow this and additional works at: https://aisel.aisnet.org/pacis2018

Recommended Citation
https://aisel.aisnet.org/pacis2018/75

This material is brought to you by the Pacific Asia Conference on Information Systems (PACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in PACIS 2018 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
“Know” Your Mobile Customers: A Design Approach to An Android-Based Mobile Analytics Tool

Research-in-Progress

Qi Qi Jiang
Assistant Professor,
Department of Digitalization,
Copenhagen Business School
Copenhagen, Denmark
qj.digi@cbs.dk

Lele Kang
Assistant Professor,
School of Information Management,
Nanjing University
Nanjing, China
lelekang@nju.edu.cn

Abstract

There is no doubt that we are living in the mobile age. Massive commercial data from mobile users provide huge potential opportunities for both merchants and researchers. However, compared with Web analytics, how to analyze the mobile data is still under debate. Some companies have adopted similar measurements or metrics to those that have been used in Web analytics tools to depict mobile activities. However, such measurements might bias the interpretation because the behaviors on mobile devices are fundamentally different from behaviors in an online context. Hence, we propose a prototype of an Android-based mobile analytics tool with novel mechanisms for better observing user behaviors in the mobile context, by which the individual behavioral data can be captured at the activity level. The detailed design, such as prototype structure, data structure, specific classes or methods, and sample codes, will be provided. In addition, some potential research opportunities will be elaborated. Furthermore, the practical value of our proposed prototype will be discussed.

Keywords: Mobile Application Development, Mobile Analytics, Prototype Framework, Mobile Commerce

Introduction

There is no doubt that we are living in the mobile age, using various mobile devices, such as smartphones and tablets. This mobile revolution was unveiled with the birth of the iPhone and reached a mature business model when Google published its own mobile OS (operating system), Android, to challenge the market leadership position of Apple. Through close cooperation with hardware manufacturers, Android grew to occupy 52.5 percent of the mobile market in the third quarter of 2011, based on the statistics presented by Gartner Inc. Such wide proliferation of mobile devices among consumers presents an opportunity for practitioners and researchers to better understand consumer preferences and behaviors. For example, Venkatesh et al. (2017) proposed the design and evaluation of auto-ID enabled shopping assistance artifacts for mobile users. This study seeks to develop an Android-based mobile analytics tool to accomplish this goal of studying mobile user behavior.

Such a need is evident. Particularly, we can find diverse tools for website analytics, such as Google Analytics, Yahoo! Web Analytics, Adobe Omniture, Webtrends, etc. Several industrial organizations, such as JICWEBS (The Joint Industry Committee for Web Standards in the UK and Ireland), ABCe
(Audit Bureau of Circulations electronic, UK and Europe), the DAA (Digital Analytics Association), etc., have attempted to initialize some universal definitive terms for Web analytics, such as bounce rates, session duration, and active time/engagement time, to name a few (Farris et al. 2010). However, in the mobile analytics market, the situation is a bit messy. Some tools were initially designed for mobile analytics, such as Localytics (http://www.localytics.com) and Flurry (http://www.flurry.com), while the bulk of them are considered sub-features among the Web analytics tools, such as Google Mobile Analytics (http://www.google.com/analytics/features/mobile.html) and mixpanel (https://mixpanel.com/). These tools use similar measurements or metrics to Web analytics, which may bias the interpretation because the behaviors on mobile devices are fundamentally different (Rice and Katz 2003; Hong et al. 2006). For example, Web analytics tools focus more on the traffic of page(s) (Farris et al. 2010), while mobile analytics tools pay more attention to the activities, services, or function (Xu et al. 2011), such as location-based services, photographing, communication, and socialization. In other words, the most essential feature of a qualified mobile analytics tool should depart from traffic or usage frequency to the actions or activities, which is also the primary motivation for us to propose a prototype of a mobile analytics tool.

In the next section, a comprehensive extent of reviews of mobile technological services will be listed. Such reviews provide a brief picture of the academic findings and the industrial experiences, which could serve as the vane of future commercial trends in the mobile industry. We propose our prototype of an Android-based mobile analytics tool in Section 3, in which the detailed designs, such as used built-in classes, application structures, database design, and basic mechanisms, are shown. Some potential research propositions and research designs in terms of our proposed prototype will be discussed in Section 4. Section 5 ends the article with concluding comments.

**Literature Reviews of Mobile Technological Services**

In this section, we conducted a comprehensive review of extant literature on several mobile technological services, including LBS (location-based service, socialization, and mobile games. These works afford sound evidence to show the importance of designing a function-oriented mobile analytics tool rather than a tool that serves as the mobile replica of a Web analytics tool.

*Location-Based Service (LBS)*

LBS is a network-based service that can generate information about a mobile device’s location or position, through which value-added services could be provided to users (Wang and Lin 2017, Barnes 2003). In the last decade, several studies have investigated how LBS could affect or change individual behaviors. For instance, Rao and Minakakis (2003) proposed that LBS could deliver high-quality services to end users in terms of reducing confusion and improving the user experiences; Xu and Gupta (2009) found that the privacy concern would significantly influence continued adoption of LBS; Gerpott and Berg (2010) surveyed 662 German users and found the willingness to adopt LBS was lower due to the concern of security or privacy and the underestimated usefulness; and Xu et al. (2005) found that the trust belief levels of consumers could mitigate the perception of privacy risk and increase the intention to uncover personal information for LBS usage.

These prior works consistently highlighted that LBS is a significant value-added feature for building up mobile applications. This was echoed by Gartner Inc.1, which deemed that LBS, as a part of context-aware services, would become the most disruptive technology in the next few years. Furthermore, an increasing number of companies have added the LBS feature into their products. Even the core functions of such products have limited relevance with LBS, such as WhatsApp (an instant messaging application for smartphones), BaiduInput (a typing application for smartphones) and MixZing (a mobile media player), to name a few.

---

Mobile Socialization

Several novel prototypes for mobile social networking have been proposed in previous works. For instance, Miluzzo et al. (2008) designed a mobile social application by utilizing sensor-generated data; Pietiläinen et al. (2009) designed and implemented a middleware for mobile social networking; and Lu et al. (2005) found that the adoption of mobile social networks depended on social influences, personal innovativeness, perceptual beliefs, and perceived usefulness. These works largely measured the perception of mobile socialization or the IT artefact design of mobile social application, but they did not investigate the methods to observe and understand the real behavioral activities. The proposed prototype in this study will address this gap.

Mobile Games

Mobile games is another emerging field in the mobile application industry. Several studies have sought to identify factors that influence the adoption of mobile games (Ha et al. 2007). Some researchers have attempted to integrate game elements into mobile applications for marketing campaigns or promotions. For instance, MacInnes et al. (2002) proposed a novel business model that integrates mobile gaming elements for managing customer relationships; as a subset of mobile marketing, mobile advertising games, or m-advergames, were classified into three main types for future marketing research: single-player games, multiplayer games involving small physical distance (e.g., using Bluetooth), and context-aware multiplayer games (Salo and Karjaluoto 2007). Gartner Inc. 2 predicted that more than 50 percent of organizations would implement gaming mechanisms into the innovation management process by 2015. In addition, numerous mobile applications include embedded gaming mechanisms for diverse aims, such as consumer education (Ducati Challenge, http://www.dtales.it/ducati/), brand awareness (iButterfly, http://smibutterfly.com/), or discount coupons (WikiTude, http://www.wikitude.com/).

In summary, this section documents three mobile technological services that are currently prevalent in either industries or academia. Certainly, some other services or technologies, such as mobile payment, RFID, and augmented reality, are also pervasive in our daily usage. Due to the limitation of page length, we could not elaborate on each of them in this study. However, the existing reviews show the diversity of mobile services and their departure from typical Web-based services. In light of the differences, it is particularly necessary to have a novel prototype that could capture user behaviors at the activity level.

Prototype and Design Approach

The basic mechanism of our proposed mobile analytics tool, IutService, is to monitor and log Android activities and upload such information iteratively. The overall prototype could be classified into three parts, which are illustrated in Figure 1.

![Figure 1. Overall Prototype Structure](http://www.gartner.com/newsroom/id/1629214)
monitored and recorded iteratively: “Identity,” “Top Activity,” and “Other Activities.” The term “Identity” includes two components. One is the name of application, which is retrieved in terms of the super class of the application because most of the Java programs were named using the format “com.appname.xxxx.” Hence, the name of each application could be simply obtained by parsing such a string. The other component is the unique ID of each mobile device. We obtained this by a user-defined method, getDeviceID3, whose sample code is shown in Table 1. As we discussed in the previous section, when investigating mobile behaviors, attention should be paid to more personalized activities. Hence, measuring the activities at the individual level is essential for better depicting the personal behaviors. “Top activity” denotes the current foreground activity or function, while “Other Activities” denotes the activities or functions that are concurrently running in the background. Take, for example, the mobile application of Facebook. The default login page is the newsfeed. Thus, the top activity is “browsing the newsfeeds.” However, when users switch to the “Check-in” function, the “Top Activity” will change to the LBS. The “browsing the newsfeed” activity will then be labelled as one of the “Other Activities.”

Table 1. Sample Java Codes of “getDeviceID”

```java
public static String getDeviceID(Context ct)
{
    String ID=Secure.getString(ct.getContentResolver(), Secure.ANDROID_ID);
    if(ID!=null)
        return ID;
    else
        ID="35"+Build.BOARD.length()%10+ Build.BRAND.length()%10 +
        Build.CPU_ABI.length()%10 + Build.DEVICE.length()%10 +
        Build.DISPLAY.length()%10 + Build.HOST.length()%10 +
        Build.ID.length()%10 + Build.MANUFACTURER.length()%10 +
        Build.MODEL.length()%10 + Build.PRODUCT.length()%10 +
        Build.TAGS.length()%10 + Build.TYPE.length()%10 +
        Build.USER.length()%10;

    return ID;
}
```

The second functional component is the “Iut Database.” We used the built-in database, the Android SQLite database, to store the temporal logs obtained from “IutMonitor Thread.” The logs will be uploaded to the server with the mechanism of breakpoint transmission by the last functional component, “Upload Log.” Notably, a detector was also designed for estimating whether the Wi-Fi is switched on before transmitting the stored logs, and some sample codes are shown in Table 2. If the Wi-Fi is detected, the logs will be transmitted immediately. Otherwise, a fixed extent of logs will be transmitted via mobile network by a predefined timeslot. For instance, if the fixed extent of logs and timeslot was predefined as 100 rows of records and one hour, respectively, then only 100 records would be uploaded every hour when the Wi-Fi was not available.

All the uploaded information is stored in the central background. Notably, if users use any LBS function, the detailed geographic information, such as longitude and latitude, will also be recorded. A snapshot of the data structure is displayed in Figure 2.

The first column includes the unique ID of the phone. The second and third columns contain the beginning and ending times of an activity session. We also compute the duration of such a session, which is recorded in column four, in milliseconds. The main activity is recorded in column five. The detailed function of each mobile application could be parsed in terms of these strings. If any LBS function was deployed, the longitude and latitude would be recorded in column six. For more detailed

---

3 [http://www.pocketmagic.net/2011/02/android-unique-device-id/#.UQszSB2Tw3d](http://www.pocketmagic.net/2011/02/android-unique-device-id/#.UQszSB2Tw3d)
information on the classes or methods, please refer to Table 3, which includes both the declarations and the descriptions.

<table>
<thead>
<tr>
<th>Type</th>
<th>Class</th>
<th>SuperClass</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Defined</td>
<td>IutLog</td>
<td></td>
<td>getHeader (Context ct)</td>
<td>Obtain log header</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>getHeaderUID (Context ct)</td>
<td>Obtain log header and unique ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>getHeaderHDID</td>
<td>Obtain log header and hardware ID</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>getHeaderLT</td>
<td>Obtain the type of log header</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>getHeaderStarttime</td>
<td>Obtain the beginning time of log header</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>getHeaderNsource</td>
<td>Obtain the source of log header</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>getAllMessage</td>
<td>Obtain the whole log, header plus context</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>clearContent</td>
<td>Clear the content</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>write2DB</td>
<td>Write the logs into database</td>
</tr>
</tbody>
</table>

Figure 1. Snapshot of Data Structure

Table 2. Sample Java Codes of “Wi-Fi Detector”

```java
public class IutService {
    private void setBroadcast() {
        networkBroadcastReceiver=new BroadcastReceiver() {
            @Override
            public void onReceive(Context context, Intent intent) {
                ConnectivityManager conn=(ConnectivityManager)context.getSystemService(Context.CONNECTIVITY_SERVICE);
                NetworkInfo networkInfo=conn.getActiveNetworkInfo();
                if(networkInfo!=null && networkInfo.isConnected()) {
                    if(networkInfo.getType()==ConnectivityManager.TYPE_WIFI) {
                        …
                    } else if(networkInfo.getType()==ConnectivityManager.TYPE_MOBILE) {
                        …
                    }
                }
            }
        };
    }

    // Other methods...
}
```

Table 3. Detailed Information on the Classes or Methods
<table>
<thead>
<tr>
<th>Class</th>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LogManager</td>
<td>createAppsInstance</td>
<td>Create the instance of applications</td>
</tr>
<tr>
<td></td>
<td>createWebsiteInstance</td>
<td>Create the instance of website</td>
</tr>
<tr>
<td>IutMainActivity</td>
<td>IsServiceRunning</td>
<td>Detect whether the service is running or not</td>
</tr>
<tr>
<td>IutMonitor</td>
<td>writeRecord2Log</td>
<td>Write the data into logs</td>
</tr>
<tr>
<td></td>
<td>getListRunningActivity</td>
<td>Obtain the running activity</td>
</tr>
<tr>
<td>IutService</td>
<td>setBroadcast</td>
<td>Set the broadcast</td>
</tr>
<tr>
<td></td>
<td>startMonitorThread</td>
<td>Start to monitor the thread</td>
</tr>
<tr>
<td></td>
<td>stopMonitorThread</td>
<td>Stop monitoring the thread</td>
</tr>
<tr>
<td>IutEncryptionNew</td>
<td>encodeString</td>
<td>Encoding (Des+Base64)</td>
</tr>
<tr>
<td>IutGpsManager</td>
<td>startGpsDetect</td>
<td>Start to detect GPS chip</td>
</tr>
<tr>
<td></td>
<td>endGpsDetect</td>
<td>Stop detecting GPS chip</td>
</tr>
<tr>
<td></td>
<td>setData</td>
<td>Set up the geographic data</td>
</tr>
<tr>
<td></td>
<td>getData</td>
<td>Obtain the geographic data</td>
</tr>
<tr>
<td>IutNetUtils</td>
<td>networkAvailable</td>
<td>Detect whether the WiFi is available</td>
</tr>
<tr>
<td></td>
<td>uploadRecord</td>
<td>Upload the log records</td>
</tr>
<tr>
<td>IutUniqueIdUtils</td>
<td>getDeviceID</td>
<td>Obtain the device unique ID</td>
</tr>
<tr>
<td>IutUploadManager</td>
<td>uploadLog</td>
<td>Upload the log files</td>
</tr>
<tr>
<td>Android Built-in</td>
<td>BootReceiver</td>
<td>BroadcastReceiver</td>
</tr>
<tr>
<td></td>
<td>onReceive</td>
<td>This method is called when the BroadcastReceiver is receiving an Intent broadcast.</td>
</tr>
<tr>
<td>IutMainActivity</td>
<td>Activity</td>
<td>onCreate</td>
</tr>
<tr>
<td>IutService</td>
<td>Service</td>
<td>onCreate</td>
</tr>
<tr>
<td></td>
<td>onStartCommand</td>
<td>Called by the system every time a client explicitly starts the service by calling startService(Intent), providing the arguments it supplied and a unique integer token representing the start request.</td>
</tr>
<tr>
<td>PassiveChangeReceiver</td>
<td>BroadcastReceiver</td>
<td>onReceive</td>
</tr>
<tr>
<td>IutDatabaseUtils</td>
<td>getAllLogSize</td>
<td>Obtain the size of all logs</td>
</tr>
<tr>
<td></td>
<td>deleteLog</td>
<td>Delete the logs</td>
</tr>
<tr>
<td></td>
<td>getLogBySize</td>
<td>Obtain the size of the maximum log</td>
</tr>
<tr>
<td></td>
<td>insertLog</td>
<td>Insert the log into database</td>
</tr>
<tr>
<td>dbhelper</td>
<td>SQLiteOpenHelper</td>
<td>onCreate</td>
</tr>
<tr>
<td></td>
<td>onUpgrade</td>
<td>Called when the database needs to be upgraded.</td>
</tr>
</tbody>
</table>
Research Proposition and Future Work

Our proposed prototype of a mobile analytics tool could grant an opportunity for two streams of research: a mobile field experiment and human behavioral analytics. Experimental researchers could implement the structure of the proposed mobile analytics tool into their own applications. By collecting such activity data, the researchers could better understand their subjects’ behaviors. Researchers who investigate behavioral analytics could revise our proposed prototype to enable it to observe the overall activities across different apps. Prior literature (Sismeiro and Bucklin 2004) investigating Web analytics has depicted that cross-website observation is essential for understanding online consumer behaviors. In a similar vein, the revised prototype could achieve the function of observing cross-application activities as well.

For future research, three actions could be taken. First, a Web-based interface ought to be developed for the users. The operators or owners of some mobile applications could simply log into the system to get the usage information of their mobile applications. Second, open APIs should be publicized in order to encourage more innovative ideas to be contributed. Third, some generalizable classes should be created for the customized requirement. Different industrial fields may have different analytical insights or needs for their users or customers. Hence, different algorithms or functions should be provided for such customized needs.

Conclusion

In this article, we first provide a brief picture of the existing mobile analytics tools market and found the shortages or limitations in the current stage. Next, a comprehensive literature review of mobile technological services is provided, from which we were able to find the necessity of observing user behaviors by individual activity or function. Such findings motivate us to propose an activity-oriented mobile analytics tool. The detailed design approach and sample codes are provided. We also list the main Java classes or methods that make up the application. Some potential research propositions and future works are also discussed.

Acknowledgements

The study was supported by the following grants: the National Natural Science Foundation of China (NSFC 71704078, NSFC 71702133, NSFC 71532015, NSFC 71771177, NSFC 71771179), Jiangsu Social Science Foundation (16TQC002), the Innovation Program of Shanghai Municipal Education Commission (15CG20), Fundamental Research Funds for the Central Universities and the Research funds of Nanjing University (010814380002), the Shanghai Pujiang Program (16PJIC086).

References