Android OS: A Review

Przemyslaw Gilski[1], Jacek Stefanski[1]

[1]*Gdansk University of Technology, Faculty of Electronics,Telecommunications and Informatics, Department of Radio Communication Systems and Networks, ul. Gabriela Narutowicza 11/12, 80-233 Gdansk, Poland*

***Abstract* – Due to the technological development, mobile terminals have evolved into functionally sophisticated devices such as smartphones and tablets. The Android platform has become one of the most popular operating system with millions of new users each year. Despite many studies, none have provided a comprehensive description of this operating system. In this paper we present a review of the Android OS. We describe the platforms history including improvements involved in each release as well as the systems architecture and project structure.**

***Keywords* – Android OS, Mobile computing, Operating systems, Software.**

1. Introduction

As handheld devices such as smartphones and tablets become more popular, the role of an operating system grows significantly. Android OS powered devices are full of integrated hardware such as cameras, gyroscopes, accelerometers, WiFi and Bluetooth wireless communication modules, and of course a touch screen. Most of them contain built-in GPS (*Global Positioning System*) and GLONASS (*GLObalNAvigation Satellite System*) receiver modules. Like all operating systems, Android enables applications to make use of those hardware features and provides a suitable environment.

2. Overview of Android OS

The platform was originally created by Android Inc., which was then later bought by Google and released as the AOSP (*Android Open Source Project*) in 2007. This announcement was accompanied by the founding of the OHA (*Open Handset Alliance*), a consortium dedicated to develop and distribute Android. The software has been released under the Apache license as a free open source license. Thanks to the rapid development, a new major release of Android takes place every few months [1][2].

The OHA is a group of several hardware, software and telecom companies including Google, Intel, NVIDIA, Qualcomm, Motorola, HTC and T-Mobile, for which Android is the flagship software. Their main goal is to develop technologies that will significantly lower the time and cost of developing and distributing mobile devices and services. The Android platform, being a complete set of software, that is an operating system, middleware and key mobile applications, is the first step in this direction.

Android is a fairly young platform, the development takes place very qu ickly. Each major release is named in alphabetic order after a dessert or sugary treat. Table 1.and Fig. 1. represent the percentage of active Android devices with access to Google Play.

Table 1.Active Android devices with access to Google Play

Version	Name	Distribution[%]
2.2	Froyo	0.8
2.3	Gingerbread	14.9
4.0	Ice Cream Sandwich	12.3
4.1-4.3	Jelly Bean	58.4
4.4	KitKat	13.6

Data were collected by Google during a 7-day period ending on June 4, 2014. Any version with less than 0.1% distribution are not shown.

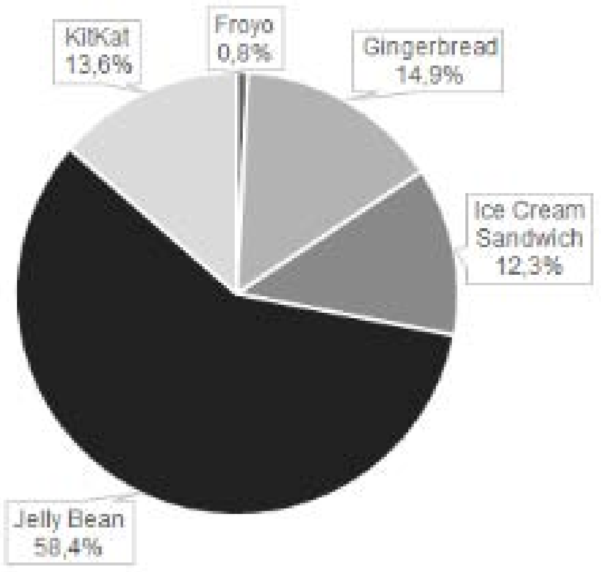


Figure 1.Distribution of Android OS

Google Play, formerly known as Android Market, is a digital application distribution platform for Android and an online electronics store developed and maintained by Google. The service allows users to browse and download music, books, magazines and applications published through Google.

According to presented data, the most common version of Android is still Jelly Bean with over 58% distribution, whereas the latest KitKat is present in more than 13% of handheld devices. Table 2.presents the Android release history.

Table 2.Android release history

Version	Name	Release Date
1.5	Cupcake	April 2009
1.6	Donut	September 2009
2.0-2.1	Éclair	October 2009
2.2-2.2.3	Froyo	May 2010
2.3-2.3.7	Gingerbread	December 2010
3.0-3.2.6	Honeycomb	February 2011
4.0-4.0.4	Ice Cream Sandwich	October 2011
4.1-4.3.1	Jelly Bean	June 2012
4.4-4.4.4	KitKat	October 2013
5.0-5.0.2	Lollipop	November 2014

Major improvements in each release involve:

- Cupcake – UI (*User Interface*) refinement updates of all core elements, accelerometer-based application rotations, on-screen soft keyboard, video recording and playback, stereo Bluetooth support;
- Donut – quick search box, VPN (*Virtual Private Network*) and 802.1x support, battery usage indicator;
- Éclair – Bluetooth 2.1 support, additional camera modes, multiple e-mail and account support;
- Froyo – tethering and WiFi hotspot capability, JIT (*Just In Time*) compiler, Adobe Flash support;
- Gingerbread – NFC (*Near Field Communications*), additional sensor support (gyroscope, rotation vector, linear acceleration, gravity, barometer), multiple camera support, large screen resolution support (tablets), Google Talk;
- Honeycomb – tablet-only android update, connectivity for USB (*Universal Serial Bus*) accessories, high-performance WiFi lock;
- Ice Cream Sandwich – WiFi Direct, Face Unlock, numerous improvements (stability, optimization, screen rotation, graphics);
- Jelly Bean – Google Wallet, Google Now, USB audio, Photo Sphere panorama photos, multiple user accounts, Miracast wireless display support;
- KitKat – NFC host card emulation, new experimental runtime virtual machine (ART), Bluetooth MAP (*Message Access Profile*) support;
- Lollipop – Dalvik replaced with ART with AOT (Ahead-Of-Time) compilation, support for 64-bit CPUs, OpenGL ES 3.1 support, recent activities screen with tasks instead of applications, project Volta (battery life improvements), audio input and output through USB devices.

3. Android OS Architecture

Unlike other mobile operating systems such as Windows Phone or iOS, Android applications are written in Java and run in a Dalvik VM (*Virtual Machine*). This virtual machine is a core component, because all Android applications and the application framework are executed by it. Similar to other platforms, applications can be obtained from a dedicated place called Google Play [3].

With Android 5.0 (Lollipop), Google made internal changes to the platform, with the Android Runtime officially replacing Dalvik for improved application performance. Despite this fact, devices utilizing Dalvik are still in majority.

Principally, Android consists of a UNIX-like operating system based on a 2.6 Linux kernel. The platform has been of course enriched with all necessary elements in order to provide basic functions including network connectivity like GSM and UMTS cellular systems.

The Android system architecture shown in Fig. 2.consists of five layers: Linux kernel, libraries, Android runtime, application framework and applications [4].

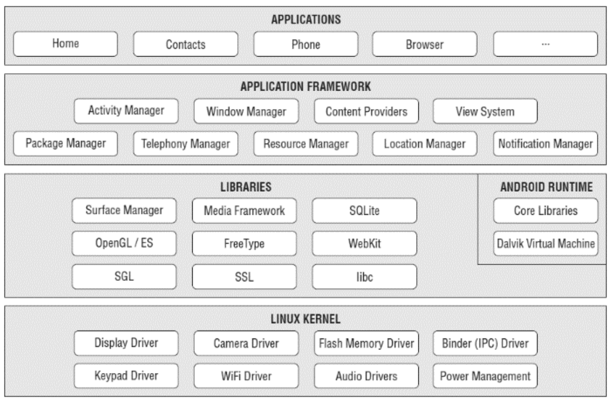


Figure 2.Architecture of Android OS

The kernel is a 2.6 series Linux kernel device driver modified mainly for power and memory management purposes.

The next level consists of native libraries written in C or C++. Due to the fact that Android was designed to run on low powered CPU (*Central Processing Unit*) and GPU (*Graphics Processing Unit*) devices with limited memory, the main libraries like *libc* or *libm* were developed to ensure low memory consumption. This layer contains also a Surface Manager responsible for screen access, a Media Framework optimized for handling audio and video codecs, e.g. MP3, MPEG-4 (*Moving Picture Experts Group*), H.264, a SQL (*Structured Query Language*) database and a native web browser engine (WebKit).

The Android Runtime includes the Dalvik Virtual Machine and Java Core Libraries. This VM is an interpreter for byte codes that have been converted from standard Java *jar* files into *dex* files, which are more compact and efficient than class files considering the limited memory and battery power of an Android mobile device.

The most important part of the Application Framework layer is the Activity Manager, responsible for controlling the life cycle of applications. Any Android application runs in its own sandboxed Dalvik VM and can consist of multiple components, e.g. services, activities, content providers, which can interact with each other within one single or many different applications on demand.

Additionally to the actual Java class library, the Android SDK contains all tools necessary to build an application.

- AAPT *(Android Asset Packaging Tool*) – enables the developer to create, view and update *zip* archives in the form of *apk* files. Those files, containing all the resources and the program itself, can be transferred to and installed on any Android device or emulator;
- ADB (*Android Debug Bridge*) – sets up a connection with a physical Android device or emulator in order to transfer and install *apk* files on it. It also enables the developer to execute remote shell commands;
- AIDL (*Android Interface Definition Language*) – used to generate code enabling two processes on a single Android device to communicate with each other using IPC (*InterProcess Communication*);
- DDMS (*Dalvik Debug Monitor Service*) – provides port-forwarding services, screen capturing, thread and heap information on the device, logcat, process and radio state information, incoming calls, SMS (*Short Message Service*) and location data indication;
- DX (*Dalvik cross-assembler*) – converts byte code *class* files into binary *dex* files executed by Dalvik VM.

The SDK also contains an emulator which simulates all the functionality of a real Android device, as shown in Fig. 3.

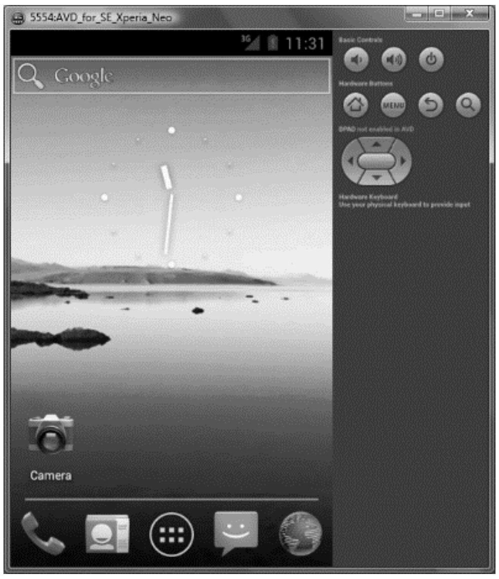


Figure 3.Android emulator

This is achieved by booting a system image representing a virtual handheld device running Android OS.

4. Dalvik VM

Despite the fact that all Android applications are written in Java, Android uses its own virtual machine instead of a standard Java VM. The necessary byte code interpreter is called Dalvik. Instead of using a standard byte code, Dalvik has its own byte code format adjusted to the needs of Android powered devices. This makes the byte code more compact than standard Java byte code.

As a multitasking operating system, Android allows every application to be multithreaded and spread over multiple processes. On account of improved stability and enhanced security each application is separated from other running applications and runs in a sandbox environment in its own Dalvik VM instance. This requires Dalvik to be small and add only little overhead [5].

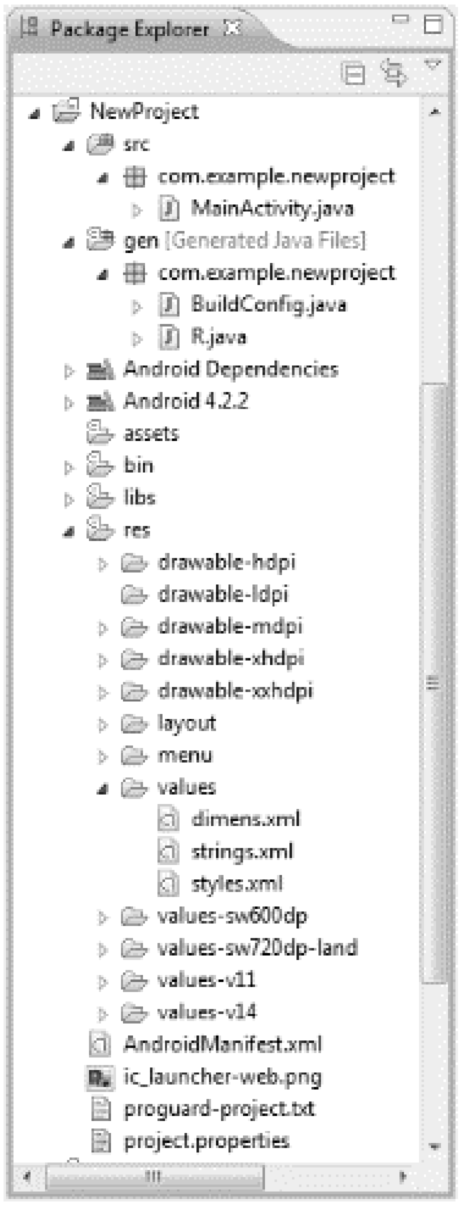
Java applications for Dalvik are compiled the same way as other Java programs. However, instead of compressing and packaging the resulting *class* files into a *jar* file, they are translated into a *dex* file by the DX tool. These files include the Dalvik byte code of all Java classes of the application. Together with other resources like images, sound files or libraries, *dex* files are packed into *apk* files.

In order to save storage, *dex* files only contain unique data, e.g. if multiple class files share the same string, this string would only appear once in the *dex* file and the multiple occurrences would be represented by pointers to this one string. The same mechanism is used for constants, method names and objects.

The Dalvik byte code is also designed to reduce the number of necessary read and write operations. For this purpose Dalvik uses its own instruction set with a fixed 16-bit length. Furthermore, Dalvik is a register-based VM which reduces the number of required code units and instructions. The given register width is equal to 32 bits, allowing each register to hold two instructions. Instructions shorter than 16 bits are zero-filled [6].

5. Android Project Structure

The Android build system, alike other Java projects, is organized around a specific directory tree structure. However, some specifications are fairly unique to this platform, but their main objective is to make software development much easier in order to prepare the actual application that will run on a device or emulator [7][8].



```
Package Explorer
NewProject
  src
    com.example.newproject
      MainActivity.java
  gen [Generated Java Files]
    com.example.newproject
      BuildConfig.java
      R.java
  Android Dependencies
  Android 4.2.2
  assets
  bin
  libs
  res
    drawable-hdpi
    drawable-ldpi
    drawable-mdpi
    drawable-xhdpi
    drawable-xxhdpi
    layout
    menu
    values
      dimens.xml
      strings.xml
      styles.xml
    values-sw600dp
    values-sw720dp-land
    values-v11
    values-v14
  AndroidManifest.xml
  ic_launcher-web.png
  proguard-project.txt
  project.properties
```

Figure 4.Android project structure

Each Android project consists of key elements included in the root directory, as shown in Fig. 4:

- *src* – contains the source code of an application;
- *gen* – holds a compiler-generated file that references all sources found in the project;
- *assets* – encloses other static files used for deployment;
- *res* – the parent folder of drawable, layout, values and AndroidManifest, including all resources used in the application, such as icons and GUI (*Graphical User Interface*) layout;
- *drawable* – contains images or image-description files supporting devices with different screen resolution;
- *layout* – holds a XML (*eXtensible Markup Language*) file describing the appearance of an application;
- *values* – encloses other resources used by the application, including strings, arrays, styles and colors;
- *AndroidManifest.xml* – defines activities, content providers, services and intent receivers, also specifies permissions required by the designed application.

Although the size and complexity of an Android project can vary, the structure will present itself similarly. When installed, all applications are visible in the device main menu, so-called launcher, in the form of icons.

6. Conclusions

The wide support from large companies, especially Google, have made Android one of the most important contestants in the mobile sector. The widespread and availability of smartphones and tablets allow manufacturers to modify the system in order to fit their needs, including both hardware and software layers.

However, the inconvenient aspect of this platform, which is fragmentation, still remains. It takes a while for manufacturers to adopt a new version of Android to already released devices available on the market. Usually they do not provide a continuous support for all.

Despite all difficulties, the release of a new version takes place roughly once a year. The development itself may broaden as more hardware and software companies get involved in the project. It is worth mentioning, that the platform is well supported by people outside the main Android project.

References

[1]. Android Developers webpage, http://developer.android.com/index.html [access: 02.2015].

[2]. Schmidt H. G., Raddatz K., Schmidt A.D., Camtepe A., Albayrak S. (2009). *Google Android – A Comprehensive Introduction*. TUB-DAI.

[3]. Wang C., Duan W. Ma J., Wang C. (2011). *The Research of Android System Architecture and Application Programming*. ICCSNT.

[4]. Macario G., Torchiano M., Violante M. (2009). *An In-Vehicle Infotainment Software Architecture Based on Google Android*.SIES.

[5]. Wei T.E., Jeng A.B., Lee H.M., Chen C.H., Tien C.W. (2012) *Android Privacy*. ICMLC.

[6]. Kundu T.K., Paul K. (2010) *Android on Mobile Devices: An Energy Perspective*, CIT.

[7]. Lee W.M. (2011). *Beginning Android Application Development*. Wiley Publishing.

[8]. Murphy M. L. (2008) *The Busy Coder's Guide to Android Development*. CommonsWare.

Corresponding author: Przemyslaw Gilski
Institution: Gdansk University of Technology, Faculty of Electronics, Telecommunications and Informatics, Department of Radio Communication Systems and Networks, Gdansk, Poland
E-mail: pgilski@eti.pg.gda.pl