A Comparative Study of Mobile Phone's Operating Systems

Gaurav Jindal
Assistant Professor,
Gitarattan International
Business School, Rohini

Mayank Jain
Student (MCA),
Gitarattan International
Business School, Rohini

ABSTRACT

"Without continual growth and progress, such words as improvement, achievement, and success have no meaning." -*Benjamin Franklin*

Mobile industries seem to follow this quote very well. A continuous growth and progress have become motto of these industries. New technologies are being developed every other day to mark the continuous growth of industry. Every company is willing to provide new features and easy to use interface to their customers. But perfection is a thing which comes with time. This paper will include various features, advantages, lacking of major mobile operating systems which includes iOS by Apple, Android by Google and Symbian of Nokia. With the analysis, I have found that once a major market share holder Symbian is now on a verge to be history, while new operating systems are at a boom due to new technology and features, iOS has still been able to conserve its market share with frequent updations.

KEYWORDS: Android, Symbian, iOS, Mobile Operating System

I. INTRODUCTION

With increasing craze of mobile phones in customers, we often see a wave of confusion for selection of best phone in their minds. Hundreds of brands with different operating systems, providing tons of features to customers seems to be a mind boggling market.

"A mind-boggling bazaar of competing manufacturers and overlapping technologies"

-*William D. Marbach*

Competition in mobile industry is increasing day by day. Every mobile company wants to provide best features in their mobiles. As a result, we see various mobile companies provide with different mobile operating systems, having different features on a competitive edge. In this research, I will be talking about various mobile operating systems, together with their features, advantages and lack points. Through this paper, I will compare between these operating systems and will try to provide ideas for various new features which may be added to them so as to make them better for customers.

II. LITERATURE SURVEY

Gen.	Developed Year	Developed By	Type	Frequency	Standards	Services	Data Speed
0G	1940	-	Voice	Analogue VHF (35-44 MHz), VHF (152-158 MHz) & UHF (454-460 MHz)	-	Voice call	n/a
1G	1979	Nippon Telegraph and Telephone, Japan	Voice	Digital 150– 900 MHz	NMT, AMPS, Hicap, CDPD	Voice call	600 - 1200 bps
2G, 2.5G, 2.75G	1991	RadioLinja (Elisa Oyj), Finland	Voice & Data	Digital 400 & 450 MHz, 900 & 1800 MHz or	GSM, GPRS, EDGE, HSCSD, iDEN, DAMPS, IS-95, PDC,	Voice Call SMS, WAP,	9.6, 56 or 236.8 kbps 3G

				850 & 1900 MHz	PHS, WiDEN, CDMA2000	MMS	
3G, 3.5G	2001	NTT Docomo, Japan	Voice & Data	Digital 400 & 450 MHz, 900 & 1800 MHz or 850 & 1900 MHz	UMTS, HSPDA,WCDMA, FOMA, 1xEV-DO/IS-856, TDSCDMA, GAN/UMA, HSUPA	Voice Call SMS, WAP, MMS	384 kbps, 1.8 or 3.6 Mbps,14 Mbps
4G				Digital 400 & 450 MHz, 900 & 1800 MHz or 850 & 1900 MHz	3GPP WiMax, WiBro	Voice Call SMS, WAP, MMS	Up to 100 Mbps

Table 1: Brief History about mobiles [1]

Hexagonal cells which are used by mobile phone stations are invented by bell labs engineers. During the World War II radio phones are initially used. In 1930s it is possible to make call by a telephone customer on ship. Such types of call are very costly. During 1940s Motorola developed a two way Walkie-Talkie and a two way radio for military which is very big in size.

What is an Operating System?

An OS is the most critical software element on any running processor-based device[2]. The OS manages the hardware and software resources within a device and performs and manages basic tasks such as the recognition of input from the device keyboard and generation of output to the device's screen. It also ensures that different programs running at the same time do not interfere with each other. It is responsible for the management of memory and for communication within the device. OSs may be extended to add additional complexity and hence functionality to the code.

In the mobile world, the more complex OSs will contain, for example, UI (User Interface) elements as these become increasingly important as the devices become more complex. The OS is purposely hidden from the user who, as a general rule, will have no direct interaction with it. It is, rather, a base onto which the applications required by the user are loaded.

The OS is not only a key element in terms of the tasks it performs but the choice of OS will constrain or enable the functionality of the end device in two key respects; firstly that which is technically possible with any given OS and secondly that which is available, i.e. what applications have been developed for that OS.

The OS, provides a software platform on top of which other application programs can run. The application programs have to be written for a particular OS so the choice of OS, therefore, determines to a great extent the applications that can be offered on the end device.

The OS also provides a consistent interface for applications, regardless of the hardware it is loaded on. Communication between the OS and the applications is done through an API (Application Program Interface) which allows a software developer to write an application for one device and have a high level of confidence that it will run on another running the same OS.

III. TYPES OF MOBILE OS PLATFORMS

The Mobile OSs [4]can be differentiated, based on the existing operating systems used by computers.

a. Real-Time Operating System (RTOS)

Real-Time Operating System which responds to inputs, immediately and generates results, instantly. This type of system is usually used to control scientific devices and similar small instruments where memory and resources are crucial and constricted. This type of devices have very limited or zero-end user utilities, so more effort goes into making the OS really memory efficient and fast (less coding), so as to minimize the execution time ,in turn saving on power as well. e.g.: 8086 etc.

b. Single user, single tasking operation system

This type of OS is better version of Real time OS, where one user can do effectively one thing at a time, which means that doing more than one thing at a time is difficult in this type of OS. For instance:

The palm OS in palm hand held computer is an example of single-task OS.

c. Single user, multi tasking operating system

It allows more than one program to run concurrently like printing, scanning, word processing etc. e.g. MS Windows and Apple's Mac OS.

d. Multi-user operating system

It allows two or more users to run programs at the same time. Some OS permit hundreds or even thousands of concurrent users. e.g. UNIX, and Main Frame OS

IV. ANDROID

Android is a computing platform designed for use in some smart phones and other devices[6]. This technology, which is owned by Google, Inc., includes an operating system, software, and applications. The operating system is based on Linux, which provides advanced computer processing. Android technology is maintained and continually developed by the Android Open Source Project (AOSP).

History of Android

Google purchased Android Inc., a 22-month-old Palo Alto, California, start-up in July 2005. Android Inc. was co-founded by Andy Rubin, maker of mobile device Danger Inc. The purchase was key in Google's move into the wireless technology market. In 2008, Google introduced the HTC Dream as the first marketed phone to use Android technology. Since that time, this platform use has expanded to other smart phones, tablet computers, E-readers, netbooks, and other devices.

Android applications

Although Android technology is increasingly being used on a range of devices, the most common hardware to use this platform is mobile phones. A large community of developers regularly write applications (apps), including games, social networking, and business modules, for Android smart phones. There are a wide range of free Android apps, including games and productivity titles, and paid apps are even more common. Android technology — which is used by thousands of developers because it is freely available for download — has given software developers the opportunity to sell their creations to a wide group of consumers.

Programming for Android

Android technology is based on Java software applications. This technology requires the use of a special software development kit (SDK) to create applications for an Android device. The SDK is freely available for download from the Internet. For this reason, and because it will work on multiple operating systems, many software developers prefer Android technology over that used in other smart phones.

Smart phones have evolved into devices that use touch screens for navigation. Android technology provides specific application programming interface (API) modules to developers that take advantage of this. The touch screen enables the user to select and scroll through information with the stroke of a finger.

What's so different in Android?

The good news is for both the consumers and developers. While consumers could enjoy a low-cost Smart phones running Android, developers were given an unrestricted customization rights[3]. From a developer's point of view, Android has several advantages, as listed below:

• The entire Application framework can be reused and replaced by selective components

• Dalvik virtual machine enhances the power management systems (Learn about Dalvik VM

in the following subtitle)

• Support for 2D and 3D graphics (OpenGL ES 1.0), So lot of business for animation developers.

• Reliable and enhanced data storage (using SQLite framework)

• Developers can create media common applications since it supports common media

file formats(MPEG, MPEG3, MPEG4, H.286, AAC, AMR, JPG, PNG, GIF and more)

• GSM, EDGE, 3G, HSCSD, Wi-Fi network applications support (Depends on hardware)

• Open source Web-Kit Engine-based web-browser

• GPS, Navigational compass, Touch-Unlock, and accelerometer applications support (Depends on hardware)

• Androids development environment includes a device emulator, debugger, performance profiling tool, and an Eclipse IDE plug-in

Reliability and security

Android is a multi-process system, in which each application (and parts of the system) runs in its own process. Most security[5] between applications and the system is enforced at the process level through standard Linux facilities, such as user and group IDs that are assigned to applications. Additional finer-grained security features are provided through a "permission" mechanism that enforces restrictions on the specific operations that a particular process can perform, and per-URI permissions for granting ad-hoc access to specific pieces of data.

As an open platform, Android allows users to load software from any developer onto a device. As with a home PC, the user must be aware of who is providing the software they are downloading and must decide whether they want to grant the application the capabilities it requests. This decision can be informed by the user's judgment of the software developer's trustworthiness, and where the software came from.

V. iOS(APPLE)

In what is widely regarded as his greatest presentation ever, Apple's Steve Jobs introduced the iPhone to the world on January 10th, 2007. In the five-plus years since then, the iPhone, iPad, and iPod Touch have literally redefined the entire world of mobile computing. That world is moving so quickly that iOS[8] is already amongst the older mobile operating systems in active development today. That certainly doesn't mean it's underpowered or underfeatured — quite the contrary. Through what can only be described as relentless and consistent improvement over the years, Apple has made iOS one of the most feature-rich and well-supported platforms on the market.

iOS 5, the system currently powering Apple's mobile devices, offers an easy-to-understand smartphone operating system to new users, a powerful platform for app developers, and a relatively un-fragmented experience across multiple devices. Perhaps the most remarkable thing about iOS is how similar the OS as it exists today is to the OS as it existed 2007, yet the number and breadth of features that Apple has baked in since then is mind boggling. Far from suffering from the "feature creep" that typically bogs down operating systems over time, iOS has managed to stay relatively snappy and is more internally consistent than anything else available today. And iOS 6 — launching on devices this fall — looks to evolve the story even further.

How did we get from a platform that began without third-party apps, multitasking, or even copy / paste support to where we are today?

History of iOS

During the original iPhone announcement, Apple touted that it ran on the same Unix core as Mac OS X and that it used many of the same tools. However, it was clear even then that while there may be some shared elements between OS X and this new phone OS, it was a different-enough beast to warrant its own branding. When the original iPhone launched, the OS was called "iPhone OS" and it kept that name for four years, only changing to iOS with the release of iOS 4 in June of 2010. For the sake of simplicity (and because it's a much-less awkward phrase).

iOS Applications

The next "finally" moment for iOS came in July of 2008, when Apple introduced the App Store to iOS. 3rd party apps for smartphones were the furthest thing from new, but Apple managed to make them feel that way with its system for developing, browsing, and installing them.

The App Store.

Critically, the App Store existed both on the device itself and within iTunes, where users could easily browse and install apps. This was a huge change from how mobile apps were distributed before — primarily over the web or via 3rd party app stores that were poorly integrated (if at all) into the device. Just as importantly, the App Store used Apple's already established base of iTunes music customers, so users wouldn't have to re-enter their credit card information in order to make purchases. It meant that finding and installing apps was easier than ever before and they quickly would become impulse buys.

The iOS SDK.

The second innovation was simply that the iPhone was a powerful device and Apple provided a development kit for iOS that offered incredible tools for developers. 3D games became the norm, and in general, iOS apps were more functional, better looking, and more advanced than on any other platform. The combination gave the platform a lead on apps that other companies are still trying to close in on.

"The App Store used Apple's already established base of iTunes music customers"

The introduction of apps and the App Store was not without some controversy, however. Apple did not completely open up iOS, but instead prevented users from "sideloading" any app they'd like. The only legitimized way to install apps was via the App Store, and Apple set a policy of curating apps that would and wouldn't be allowed in. Some of the rules were fairly straightforward — no porn — but others put Apple in a gray area when it came to users' desires. Apple regularly rejects certain classes of apps that are allowed on other platforms, including apps that allow tethering your computer to your iPhone for internet access.

Another, perhaps unforeseen, consequence of the App Store was that apps became much much less expensive. This has mainly been a net win for users and developers, but it did cause plenty of consternation as the price of a top-shelf mobile app rapidly dropped from the $40 range to the $5 or even 99-cent range. Top-selling charts for apps began to look like the top-40 Billboard charts for music: if a developer could find a way to the top, he or she could make big money, but it was difficult at the bottom. Most of these concerns have gone away in the last couple of years and now there are many, many development houses and independent developers making their living by selling iOS apps.

Reliability and security

iPhone has no security[5] software and Apple doesn't let people load third-party programs on the device, which could reduce the risk of infection from malicious software. When the iPhone is connected to the Web, dangerous possibilities emerge.

The iPhone Auto-Lock disables the device's screen after a preset time period of non-use, but the Passcode Lock feature takes that a step further. Whenever the device's display locks, whether due to Auto-Lock or because you've hit the iPhone Sleep button–found on the top right of the device–Passcode Lock requires a four-digit code to be entered before the device can be employed again.

The iPhone OS security APIs are located in the Core Services layer of the operating system and are based on services in the Core OS (kernel) layer of the operating system. Applications on the iPhone call the security services APIs directly rather than going through the Cocoa Touch orMedia layers. Networking applications can also access secure networking functions through the CFNetwork API, which is also located in the Core Services layer.

The iPhone OS security implementation includes a daemon called the Security Server that implements several security protocols, such as access to keychain items and root certificate trust management.

The Security Server has no public API. Instead, applications use the Keychain Services API and the Certificate, Key, and Trust services API, which in turn communicate with the Security Server. Because, unlike the Mac OS X security services, the iPhone OS security services do not provide an authentication interface, there is no need for the Security Server to have a user interface. Although Mac OS X includes a low-level command-line interface to the OpenSSL open-source cryptography toolkit, this interface is not available on the iPhone OS. For iPhone OS development, use the CFNetwork API for secure networking and the Certificate, Key, and Trust Services API for cryptographic services.

VI. SYMBION

Symbian is a mobile operating system (OS) and computing platform designed for smartphones and currently maintained by Accenture. The Symbian platform is the successor to Symbian OS and Nokia Series 60; unlike Symbian OS, which needed an additional user interface system, Symbian includes a user interface component based on S60 5th Edition. The latest version, Symbian3, was officially released in Q4 2010, first used in the Nokia N8. In May 2011 an update, Symbian Anna, was officially announced, followed by Nokia Belle (previously Symbian Belle) in August 2011. *Symbian OS* was originally developed by Symbian Ltd. It is a descendant of Psion's EPOC and runs exclusively on ARM processors, although an unreleased x86 port existed. Some estimates indicate that the number of mobile devices shipped with the Symbian OS up to the end of Q2 2010 is 385 million.

By 5 April 2011, Nokia released Symbian under a new license and converted to a proprietary model as opposed to an open source project.

On 11 February 2011, Nokia announced that it would migrate from Symbian to Windows Phone 7. Nokia CEO Stephen Elop announced Nokia's first Windows phones at Nokia World 2011: the Lumia 800 and Lumia 710. These phones were launched on 14 November 2011. On 22 June 2011 Nokia made an agreement with Accenture for an outsourcing program. Accenture will provide Symbian-based software development and support services to Nokia through 2016; about 2,800 Nokia employees became Accenture employees as of October 2011. The transfer was completed on 30 September 2011.

The Symbian platform was created by merging and integrating software assets contributed by Nokia, NTT DoCoMo, Sony Ericsson and Symbian Ltd., including Symbian OS assets at its core, the S60 platform, and parts of the UIQ and MOAP(S) user interfaces.

In December 2008, Nokia bought Symbian Ltd., the company behind Symbian OS; consequently, Nokia became the major contributor to Symbian's code, since it then possessed the development resources for both the Symbian OS core and the user interface. Since then Nokia has been maintaining its own code repository for the platform development, regularly releasing its development to the public repository. Symbian was intended to be developed by a community led by the Symbian Foundation, which was first announced in June 2008 and which officially launched in April 2009. Its objective was to publish the source code for the entire Symbian platform under the OSI- and FSF-approved Eclipse Public License (EPL). The code was published under EPL on 4 February 2010; Symbian Foundation reported this event to be the largest codebase transitioned to Open Source in history.

However, some important components within Symbian OS were licensed from third parties, which prevented the foundation from publishing the full source under EPL immediately; instead much of the source was published under a more restrictive Symbian Foundation License (SFL) and access to the full source code was limited to member companies only, although membership was open to any organisation.

In November 2010, the Symbian Foundation announced that due to a lack of support from funding members, it would transition to a licensing-only organisation; Nokia announced it would take over the stewardship of the Symbian platform. Symbian Foundation will remain the trademark holder and licensing entity and will only have non-executive directors involved.

VII. RESULT & DISCUSSIONS

Having a support of world's largest App store, iOS enjoys a large variety of functionalities needed in day to day life! making it easy to opt for option for the customers. On other hand, Android has a faster growing App store and is already having largest App stores to provide conglomeration in routine apps. Symbian again lacked a good support for applications thus creating another drawback for it.

Talking about user convenience, iOS supports a super user friendly GUI, making it a prominent choice among the users. Though Android is not much far behind. Whereas, Symbian suffers with an outdated GUI. As we talk about the hardware support, Apple celebrates to be the most trusted brand for hardware support. Its devices provides a best running experience with full hardware acceleration. Android, on other hand, still struggling for a better hardware support. Symbian though also provides a great platform for different applications in terms of hardware support.

Being an end user customer, price is the main concern for every person. Apple costs far more in terms of price as compared to android and symbian, thus restricting its customers to Upper or upper middle class families. Where as, android have a series of versions, supported both by a low cost as well as high end mobiles. Making it more approachable among all sections of society. Symbion on other hand has been a first choice for a variety of sections due to its seller, Nokia, one of the most trusted brands in mobiles. But now things are changing. Android is growing very fast to become the heartbeat of mobile users.

VIII. CONCLUSION

With the completion of this paper, we can conclude that every operating system has been developed by keeping in mind the targeted customers. Every Operating System provides competitive and unique features for their customers. However, iOS came out to be ever enhancing operating system with a great evolution chart over the years. On other hand, Android, being an open source operating system enjoys addition of new ideas every day by various Android lovers. Whereas, Symbian lacked the continuous updations as compared to other systems. thus creating a large loophole for Symbian to match the latest advancements.

REFERENCES

[1] Kamboj, Gupta, (2012) "Mobile Operating Systems", International Journal of Engineering Innovation & Research, Volume 1, Issue 2, ISSN: 2277 – 5668, Pp 115-120

[2] Open Source OS - The Future for Mobile? By Juniper Research

[3] Android by 2012, A study on present and future of Google's Android By Dotcom Infoway

[4] White paper on Mobile OS and efforts towards open standards By Dotcom Infoway

[5] Zhang. J. (2010), "Android vs iPhone"

[6] http://www.wisegeek.com/what-is- android-technology.htm

[7] http://developer.android.com/about/versions/jelly-bean.html

[8] http://www.theverge.com/2011/12/13/ 2612736 /ios-history-iphone-ipad