



**RESEARCH PAPER  
ANALYSES OF INTELLIGENT MOBILE DEVICE CHARACTERISTICS  
(SMART PHONES)**

---



**SMART PHONES**

---

**December 2008**

© Copyright December 2008 by Belatrix Software Factory BSF S.A.

[www.belatrixsf.com](http://www.belatrixsf.com)

[businessdevelopment@belatrixsf.com](mailto:businessdevelopment@belatrixsf.com)

+1 (617) 608-1413

## Index

<b>1. Introduction</b>	<b>3</b>
1.1 About Belatrix Software Factory	3
<b>2. Market Share</b>	<b>3</b>
2.1 Market share according to sales	4
2.2 US Market behavior in 2008	5
<b>3. Operating systems for mobile devices</b>	<b>5</b>
<b>3.1 Symbian</b>	<b>5</b>
3.1.1 Developing for Symbian OS	6
<b>3.2 Mac OS (iPhone)</b>	<b>7</b>
3.2.1 Developing for iPhone	8
<b>3.3 RIM (Research in motion)</b>	<b>11</b>
3.3.1 Developing for RIM	11
<b>3.4 Windows Mobile</b>	<b>11</b>
3.4.1 Developing for Windows Mobile	12
<b>3.5 Linux</b>	<b>13</b>
3.5.1 Developing for Android	15
<b>4. Conclusion</b>	<b>17</b>

## 1. Introduction

Belatrix Software Factory, as leader in software technologies, has been building a deep and vast knowledge on mobile device platforms and has been involved in a process of research and development on these technologies to further develop this area in the company.

The present paper contains research and analyses on the main operating systems that are used with mobile devices (SmartPhones), indicating their corresponding market shares. It also includes the required technical characteristics and skills to develop applications for SmartPhones using the various available SDKs (software development kits).

### ¿What is a Smartphone?

A Smartphone or intelligent mobile device is a cell phone with special, more advanced features than those of a typical phone, often compared to the functionalities offered by a PC. However there is no formal agreement between the manufacturers on this definition. For some, it is a phone that executes a complete and identifiable operating system that provides a standard interface and a platform to develop applications. For others, it is simply a mobile phone, with some advanced features such as: e-mail, internet and/or an integrated keyboard. For the purpose of this paper, the first definition will be used, as we are interested in knowing the operating systems for each platform and the software development and publishing processes for each one of them.

### 1.1 About Belatrix Software Factory

[Belatrix Software Factory is an Argentina Outsourcing company](#) dedicated to software development outsourcing, founded in 1993 and entirely oriented to the provision of international outsourcing since the year 2001, for clients in the US, Canada and various countries in Mid Europe and Scandinavia.

The services offered are:

- [Customized software development outsourcing from Argentina.](#)
- [Software testing and quality assurance outsourcing from Argentina.](#)
- [Graphic design outsourcing from Argentina.](#)
- Language localization.
- Remote system support.

To contact Belatrix Software Factory:

e-mail: [businessdevelopment@belatrixsf.com](mailto:businessdevelopment@belatrixsf.com)

Phone: +1 (617) 608-1413 ext. 2001

Web: [www.belatrixsf.com](http://www.belatrixsf.com)

## 2. Market Share <sup>1</sup>

The world market for smartphones is dominated by Symbian with a market share of 57.1%, followed by RIM (BlackBerry) with 17.4%. Windows Mobile is third with 12% of the world

<sup>1</sup> Data to november 2008. Main sources:

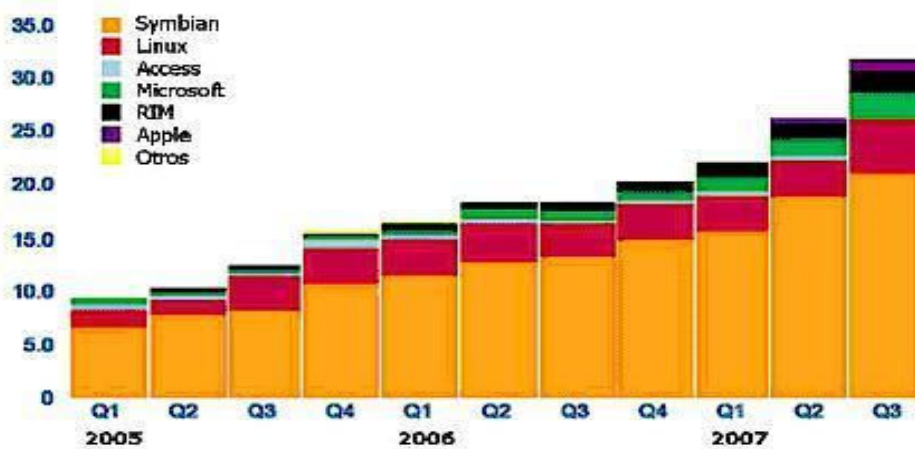
Wikipedia. ([www.wikipedia.com](http://www.wikipedia.com))

Reuters (<http://www.reuters.com/article/technologyNews/idUSTRE4A58OK20081106>).

market. The Linux based systems have a market share of 7.3%, though this number is the result of adding a number of different platforms, with great differences between them, all of which have the same operating system for the kernel. The Apple Mac OS operating system has a market share of 2.8%, but it is interesting to note that almost all the sales of iPhones (the only smartphone with Mac OS) have taken place in the US, where it is second, only exceeded by RIM.

Open Handset Alliance's Android is a recent operating system for smartphones developed by Google and T-Mobile, which was first released in the market with the HTC G1 on October 22<sup>nd</sup>, 2008 and has gained 4% of the market in less than a quarter. It is worth mentioning that the HTC G1 is sold as a "version for developers", which denotes the operating system's lack of maturity.

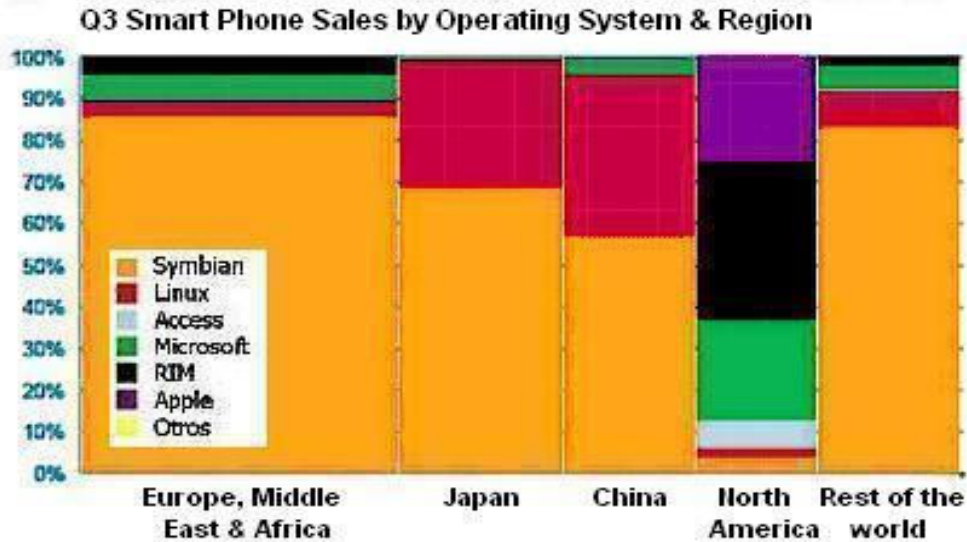
## 2.1 Market share according to sales



Note: numbers expressed as million of units sold

The US mobile market has distinct differences in its distribution when compared to the rest of the world. This can be clearly seen in the following graph (extracted from a symbian publication). Another distinct characteristic is the market of Japan and China, where there is an important number of mobiles with Unix based operating systems, which have a low market share in the rest of the world market.

The release of the iPhone 3G, between the 2nd and 3rd quarter of 2007, promoted sales for this smartphone in the US, partly due to a strong advertising campaign and partly due to the intensive use of the MultiTouch screen.



## 2.2 US Market behavior in 2008

The BlackBerry device, manufactured by Research In Motion Ltd. (RIM), increased its US multiuse phone market participation during the first quarter, while iPhone, from Apple Inc lost ground, according to a report from International Data Corporation (IDC).

According to the report, RIM's participation in the US smart phones market increased to 44.5% in the first quarter, from 35.1% in the fourth quarter the year before. Meanwhile, Apple's iPhone market share decreased to from 26.7% 19.2%, in the fourth quarter.

Motorola Inc, the main manufacturer in the US, also saw its multiuse phone sales decrease from 7.5% to 2.6%, in the previous fourth quarter.

As indicated by the IDC, Palm Inc. saw its participation increase to 13.4% from 7.9% in the preceding fourth quarter. Samsung Electronics Co Ltd's increased to 8.6% from 5.1%. High Tech Computer Corp., main manufacturer of mobiles with Windows Mobile, decreased to 4.1% from 7.9% the previous quarter, and finally, Android made its appearance in the US market with a device that is sold as "version for developers", gaining a 4% of the sales of 2008's last quarter.

## 3. Operating systems for mobile devices

Detailed below are the relevant characteristics of the operating systems for mobiles.

### 3.1 Symbian

Symbian is a proprietor operating system designed for mobile phones, with associated libraries and a user interface. It descends from Psion EPOC and runs exclusively on ARM processors. It is produced by Symbian Ltd, a society integrated by Nokia (47.9%), Ericsson (15.6%), Sony Ericsson (13.1%), Panasonic (10.5%), Siemens AG (8.4%) and Samsung (4.5%).

Symbian is a ROM based operating system designed to save battery. It is based on a

micro kernel, a minimum portion of the system has privileges on the kernel, the rest of the system runs with user privileges and it's the kernel's task to manage the interruptions and priorities. In Symbian, each application runs in its own processes and only has access to its own memory portion. This design makes Symbian applications "single thread" oriented. It is interesting to note that the system possesses components that allow multiplatform applications, i.e. different screen sizes, color, resolution, keyboards, etc. Most of these components have been designed in C++.

The design of the operating system enables Symbian devices be in constant use without the need to be reset, preserving all the user information and working correctly (proved in the lab), though the latter is being compromised due to the complexity of the latest equipments with Symbian and the multitude of programs that are external to the operating system.

Although it is not an open source software, the APIs provide public documentation and up to version 8.1 anyone could develop software for Symbian. Version 9.1 introduces a security framework and some additional features that oblige developers to digitally sign their applications. Though basic features, such as file saving, can be enabled by the user, other more sophisticated features, such as multimedia access, need certification and signature through the Symbian signature program i.e. Symbian Signed, used by independent companies or phone manufacturers used to certify. Their "TrustCenter ACS Publisher ID Certificate" is required for developers to sign their applications with the "Test House" software. The cost of the certificate is approximately u\$s200 per year. Symbian Signed only provides free certification and signature for the freeware applications through the Cellmania software.

### **3.1.1 Developing for Symbian OS**

Symbian has five user interfaces or platforms for its operating system, denominated Serie 60, Serie 80, Serie 90, UIQ and MOAP. Most of the mobiles use the Serie 60, except for the Sony Ericsson and Motorola mobiles which work under UIQ.

The native language of the Symbian OS is C++, though not in a standard implementation. There are multiple SDK's available to develop applications, the main ones being UIQ and S60. Some manufacturers offer their own SDK or SDK extensions for their products or for the family of products available to download from their web sites.

The official SDKs contain documentation, the headers, the libraries necessary to compile a Symbian software, Windows based emulators and compiler. Up to version, 8 GCC is included as compiler. From version 9 onwards it uses a new ABI (application binary interface) and it requires a different compiler.

The C++ programming for Symbian requires the use of special techniques such as descriptors or CleanupStack, which may cause relatively simple programs to become more difficult to implement than it would in other environments. Currently the programming techniques needed to develop in Symbian make most programs prone to low level routine errors rather than errors in application features.

The first official and commercial IDE for Symbian, CodeWarrior, was replaced during 2006 by Carbide C++, an IDE based on Eclipse, developed for Nokia, and it comes in three versions:

Carbide.c++ OEM Edition for device creation users.

Carbide.c++ Professional Edition for developers working with preproduction devices.

Carbide.c++ Developer Edition for application development on production phones.

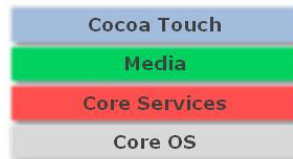
All of these versions are "free of charge" as it is shown in the "forum.Nokia.com" web page, from where you can download the IDE as well as the s60 SDK. Microsoft Visual Studio 2003 and 2005 are supported as well as development IDE, through the plug in Carbide.vs.

Many of the devices with Symbian OS as well as C++ can be programmed in OPL, Python, Visual Basic, Simkin and Perl, as well as in Java ME.

The plug in for Visual Studio, AppForge, called "AppForge CrossFire", that allowed programming a Symbian device in Visual Basic, VB.NET or C# is no longer available since 2007, when Oracle purchased the intellectual ownership of AppForge and announced that they did not plan to sell or offer support to the old products of this company. Finally there is also a Borland IDE version for Symbian, as well as tools or techniques developed by the community to program in Linux or Mac OS.

### 3.2 Mac OS (iPhone)

The iPhone/iPod Touch operating system is based, as could not be otherwise, on its older brother Mac, following the layer architecture:



*Core OS, Core Services and Media* are an exact copy of the Mac OSX source code and the only one who has suffered some modifications has been Cocoa. In Mac OSX, Cocoa was limited to the use of mouse and keyboard and for that reason it was necessary to adapt it for the use of iPhone and its touch screen. As a result it is now called Cocoa Touch.

**CORE OS:** The pillars of the iPhone operating system are:

1. OSX kernel
2. library system
3. TCP / IP stack
4. sockets
5. security
6. energy management
7. keychain
8. certificates
9. file system
10. Bonjour

**CORE SERVICES:** are the basic system services:

1. collections

2. contact directories
3. networks
4. file access
5. sqlite (data base)
6. core location
7. net services
8. threading
9. preferences
10. URL utilities

**MEDIA:** the multimedia management layer (graphics, audio, videos, etc) is composed of:

1. Audio core
2. openAL
3. sound mixer
4. sound recorder
5. video reproduction
6. management of jpg, png, tiff, pdf
7. quartz (2d)
8. Core Animation
9. OpenGL ES

#### **Cocoa Touch, the development framework for iPhone/iPod Touch:**

1. multi touch events and controls
2. accelerometer support
3. hierarchy view
4. application finder
5. alerts
6. web view
7. contact selector
8. image selector
9. camera support

#### **3.2.1 Developing for iPhone**

The iPhone operating system is the same one used by Mac for almost ten years, an over loaded Unix BSD. This operating system was developed by NeXT, a company that also developed an innovative (at that time) object oriented programming framework. That framework evolved into what we today know as Cocoa, specifically Cocoa Touch

for iPhone, which is the basis on which any native application is built for the iPhone/iPod Touch.

For the iPhone SDK to compile ARM code, which is the architecture of the real system, it is necessary to sign the application, which requires Apple certification. A new signature is necessary to distribute the application, and of course, if we want it to be available in the appStore (official site for Mac OS applications distribution), we must request Apple's permission.

The complete process to develop, compile and publish an application for iPhone is the following:<sup>2</sup>

- 1.** Register in the iPhone Dev Center, accept all the legal terms and download the SDK. Doing this we can start writing our own applications, compile them and test them in the simulator. But we still can't install them in our iPhone.
- 2.** Register in the program for developers. After completing all the forms and reading all the legal terms, we can send the request and wait for the answer on the following working day. This has a cost of U\$S 99, and we obtain the right to run our applications on our own iPhone.
- 3.** Certificate. By default an iPhone can only run applications signed by Apple and as it is not feasible to be sending them code every time we want to test something, it is necessary to create a profile with the identifier of our phone (accessible through Xcode) and input it into an Apple web form. To generate the certificate to sign our applications we will go to "Keychain Access" and under "Certificate Assistance" we will generate a new one. If everything is Ok, we can start testing our own applications on the iPhone.
- 4.** Develop! Not so long ago, developers were forbidden to talk on the SDK, which hindered the actual development process. Working with a new platform implies difficulties that normally are resolved in forums and developer groups. Luckily, Apple has abandoned this policy and it is now possible to exchange codes, knowledge, etc.
- 5.** Distribute! A special distribution certificate is necessary, which needs to be requested following the same process described above. Access KeychainAccess, request, approve, download and install. The same as in point 3, we need to create a new distribution profile in Xcode; Apple provides many data pages with all kinds of captures on how to reconfigure the Xcode project to use the distribution certificate.
- 6.** Send the application to Apple through a new web form with the application data (name, description, version...), its icon and some screen shots.
- 7.** Wait... Once sent, the application enters the review queue, without us knowing what process it is undergoing, or how much time is expected for it to end. If the application is not accepted, no matter what the reason is, there will no help on the issue. The necessary amendments can be done and the application re sent through a link used to send new versions of the same application. If the problems continue, we will have to wait a week to know if the solution has resolved the problems satisfactorily.
- 8.** Accepted!! Now is the moment to look forward to the benefits of the application facing 10 million of potential clients that may purchase it.

---

<sup>2</sup> Source: <http://www.mikeash.com/?page=pyblog/the-iphone-development-story.html>

### Developing in Cocoa Touch, development platform for iPhone<sup>3</sup>

Cocoa is an API written in a special C dialect called Objective-C. When we develop for iPhone/iPod Touch in particular, we use Cocoa Touch. The basic difference between Cocoa and Cocoa Touch is the way the user invokes commands. In Cocoa we have a physical keyboard and a mouse or trackpad, in Cocoa Touch we have a virtual keyboard and our fingers touch the screen directly.

Over the actual API, we find the tools that form part of the platform development environment.

- Xcode: is the key part of the environment. It is a very complete IDE, in which we can edit source code, access a vast volume of documentation and even find a graphic debugger. Xcode is built by many numerous Open Source modules (such as the gcc debugger and the gdb debugger).
- Instruments: is used to monitor the application being developed and tune its performance, in a wonderful graphic environment. Instruments is based on DTrace, an Open Source tool developed by Sun Microsystems. It is essential when identifying memory losses and other bugs difficult to trace.
- Dashcode: was designed to develop widgets for Mac OS X's Dashboard. The iPhone/iPod Touch SDK version is basically the same as the Mac OS X one, and it was included to aid the development of iPhone web applications.
- Simulator: allows testing the basic functionalities of the application being developed. The simulator runs an operating system similar to the real one. When it works with the simulator, Xcode compiles for x86, instead of ARM (which is the architecture of the real operating system). To compile ARM code it is necessary to sign the application, which requires Apple certification.
- Interface Builder: is basically used as user interface development (GUI), however it is much more than that. IB is where the different modules connect and it is also where many of the precompiled modules that will be seen on screen come alive.

### XCODE

Xcode is the engine that empowers the Apple integrated development environment for Mac OS X and for iPhone OS. It is also the application that is responsible for most of the project details from its initiation to implementation, allowing to:

- Create and manage projects, including specification platforms, objective requirements, dependencies and structure configurations.
- Write source code in editors with characteristics such as syntax coloring and automatic indentation.
- Browse and search project components, including header files and documentation.
- Build the project.
- Debug the project locally in the iPhone OS simulator, or remotely at source level in the graphic debugger.

Xcode builds projects from source code in C, C++, Objective-C, and Objective-C++. This produces executable files for Mac OSX, including command line tools, work frames, plug-ins, kernel extensions, bundles and applications (for iPhone OS, only

---

<sup>3</sup> Source: <http://www.iphonegurues.com/la-plataforma-de-desarrollo-cocoa-touch/>

executable applications are possible). Xcode allows customizing numerous building and debugging tools, executable packages (including property lists and found collections), build processes (including copy files, script files, and other phases of building) and a user interface (including separate code editors and multi views). It also supports various source code management systems such as CVS, Subversion and Perforce, allowing adding files to the repository, perform modifications, obtain up to date versions and compare versions.

Xcode is specially designed for development in Cocoa. When we create a project, Xcode configures its initial development environment using project templates that correspond to Cocoa project types: applications, document based applications, Core Data application, tool, collection, work frames and others. To compile Cocoa software for Mac OS X, Xcode used the GNU compiler (gcc) and to debug that software it uses the GNU source level debugger. Both gcc and gdb have been used in the Cocoa development since Cocoa was NeXTSTEP, and throughout a number of years has been refined, extended and tuned to support Cocoa binary compilation and debugging.

### 3.3 RIM (Research in motion)

BlackBerry RIM (Research In Motion) SDK is the environment to build applications for the BlackBerry wireless devices. The programming language is J2ME, which is the java platform for mobile devices. The use of these Blackberry devices is popular in the business segment and not so popular in the home and general public segments, that is why the SDK is strongly oriented to this kind of application development.

As well as the standard libraries CLDC and MIDP, RIM provides some useful libraries, such as `net.rim.device.api`, that give access to specific functions for the RIM hardware. This packet provides classes to monitor the COM ports, the radio, the keyboard and the thumb wheel. The packet `net.rim.device.api.util` includes useful builds such as ordered vectors, clonable vectors, data buffers and others. The `net.rim.device.api.system.mobitex` library is used to get data from the Mobitex network, which is the BlackBerry network.

The SDK also includes an IDE that supports debugging and the emulator to do so.

#### 3.3.1 Developing for RIM

RIM has a propriety, multitask operating system for the BlackBerry device that makes intensive use of the input devices, the keyboard and the trackball. The operating system supports MIDP 1.0 and 2.0 (as from the version 4) as well as WAP 1.2.

Any developer can write software using the BlackBerry propriety APIs, but for the application to access certain restricted functions it must be digitally signed with a certificate associated to a RIM developer account, this process guarantees the application source, but not the quality of the code.

### 3.4 Windows Mobile <sup>4</sup>

Windows Mobile is the Microsoft operating system for mobile devices. Windows Mobile evolution, which can be found in Wikipedia, reflects the pocket device evolution during the last few years. Basically created from a Windows kernel version for embedded systems it grew to an operating system for Pocket PCs (PPC) and smartphones.

---

<sup>4</sup> Source: <http://javiercancela.com/2007/10/19/introduccion-al-desarrollo-de-aplicaciones-para-telefonos-moviles-windows-mobile>

PocketPC is a Microsoft standard that sets various hardware and software requirements for the mobile devices.

Any device to be considered a PocketPC must:

- Run Microsoft Windows CE or Windows Mobile (version PocketPC) operating system
- Have a number of applications in ROM
- Include a touch sensitive screen
- Include a signaling device, called stylus
- Include a set of hardware buttons to run applications
- Be based on a processor compatible with STRONGARM

Some of the applications included with these devices are reduced versions of: Outlook, Internet Explorer, Word, Excel, Windows Media Player, etc.

Pocket PC Phone Edition, as well as the previously listed, includes radio/s for cell telephony.

A SmartPhone for Microsoft is a Pocket PC Phone Edition, generally without "touch screen".

### **3.4.1 Developing for Windows Mobile**

Microsoft provides a free SDK, "Windows Mobile 6 Professional and Standard Software Development Kits Refresh", that includes everything necessary for application development on the Windows Mobile platform, but to install it we need, as indicated in the requirements section of the download page, Microsoft Visual Studio 2005 Standard Edition or superior (Visual Studio Express Editions are not supported), which implies that to develop in WM it is necessary to acquire a Visual Studio 2005 or superior license.

Once the SDK is downloaded and installed we will have all the necessary tools to develop applications for WM6, including emulators (ROM images) for the different device screen sizes available in the market.

Two types of applications can be developed for Windows Mobile: using native code or using managed code. We call native code the C++ code that uses the Windows Mobile API directly, and managed code the one that uses the .NET Compact Framework classes with C# or VB.Net. (Windows Mobile is the only important mobile platform that does not support J2ME).

What are the differences between them? The native code is quicker and occupies less space, as well as giving access to some hardware characteristics which are inaccessible from the Compact Framework. However, in most cases, develop using managed code is the best option. The size of the executable file is losing significance and if performance is critical, the application bottle necks can be developed using native code. On the whole, developing in .NET is easier.

Developing applications for Mobile has the inconvenient of the lack of Visual Studio alternatives and the investment necessary to acquire a license. On the other hand, it is positive because of the quality of the available tools (Visual Studio, emulators, the SDK and its documentation) as well as because of the active developers community, grouped around the Microsoft development portal, MSDN.

## 3.5 Linux

In the last couple of years, many companies related to the world of mobile devices, manufacturers and software or application developers, have chosen Linux as the basis for their operating systems. Japan and China are the countries where mobile Linux is more widespread, achieving a market share of 40%. Currently, there are several initiatives adapting Linux to mobile terminals. A brief overview of the more popular Linux platforms:

### **LiMo (Linux Mobile) Foundation**

Is an alliance founded by Motorola, NEC, NTT DoCoMo, Panasonic Mobile Communications, Samsung Electronics, and Vodafone in January 2007 to develop the LiMo platform, a platform based on Linux for mobile devices. Since then, new members have joined the foundation, amongst which we can find Infineon Technologies, Mozilla, SFR and Verizon Wireless. LiMo announced at the end of March 2008 the LiMo Release 1 (basic operating system without applications which is already available in two commercial phones, the Razr 2 and the Motorola Rokr E8 de Motorola). At the beginning of 2009 Release 2 is expected, improving portability and its multimedia ability.

### **OpenMoko**

Is a project to create a platform for smartphones using free software. It uses Linux's core, together with a user graphic environment built with the X.Org server, the GTK+ tool kit and the Matchbox window manager. It is based on the OpenEmbedded framework and ipkg packet systems. OpenMoko was announced in 2006 by its founders: First International Computer (FIC). The first smartphone in which OpenMoko works is the Neo1973, manufactured by FIC.

### **MOTOMAGX**

Is an operating system for mobiles developed by Motorola. This system is a combination of Linux and Java. The MOTOMAGX platform has a modular architecture that includes the following components:

- Linux operating system. This operating system adds packets from the open source community and additional components to satisfy the mobile device requirements. The first products based on MOTOMAGX are the MOTOROKRTM Z6, MOTORAZR2 V8, MOTOTM U9 and the recently announced MOTOROKR E8.
- Library platforms and middleware services. Provides platform services and manages the application life cycle, the application interactions and the platform security.
- Development environment (Java ME, WebUI, native Linux). The application environments include the necessary APIs and services.
- Applications. The applications provide the user with experience through the interactions with the rest of the components.

### **Access Linux Platform (ALP)**

It is an operating system based on Linux for smartphones capable of running PALM OS applications. This platform belongs to Access, who acquired the Palm rights in

2005. In the 2008 summer, the mobile operator Orange had announced the release of their first commercial device, the Samsung i800, project which was finally suspended.

### **ARM Linux Mobile Platform**

ARM and six companies more, amongst which we find Texas Instruments, Samsung, Marvell and Mozilla, are building a standard software platform based on Linux open source specially designed for mobile devices. The objective is to present the first Linux devices during 2009.

### **Qtopia**

Is an application platform for mobile devices that uses Linux as operating system, developed by Trolltech. There are two categories of Qtopia, one is free under a GPL (Opie) license, and the other is commercial; as well as two versions, one for mobile phones and one for PDAs. Qtopia is installed in numerous mobile devices from Sharp Corporation in the Zaurus products line, which includes more than ten different models. Archos also includes it in the PMA430, which is a multimedia device. The version for mobile phones is expected to be available in numerous devices.

### **Android**

Is a Linux based software platform for mobile devices that includes operating system, middleware and key applications. This platform is being developed by Google and Open Handset Alliance. It allows developers to write Java code using software libraries developed by Google, but it does not support programs in native code.

The Android platform was announced in November 2007 with the Open Handset Alliance, a consortium grouping 34 hardware, software and telecommunications companies dedicated to creating open standards for mobile devices. Android provides a complete software packet at all levels:

- A Linux kernel that serves as a base for the software stack and manages the basic system functions: drivers, security, communications, etc.
- A low level library layer in C and C++.
- A framework to develop applications, divided into subsystems to work as a "package manager", manage host phone hardware ("telephony manager") or access sophisticated geolocalization or XMPP messaging APIs. It also includes a screen system to manage the application user interface, visualizing a map or html rendered directly in the application graphic interface.
- Suite of applications (browser, directory, phone management).

The Android applications are programmed in Java, but they do not run on Java ME, instead, they run on Dalvik, a Java virtual machine expressly developed by Google and optimized for embedded devices.

Android allows writing managed code in a language similar to Java that uses java libraries developed by Google, but does not support developing in native code. Almost all the Android platform was released in 2008 under the "Apache free-software and open-source" license. The first phone to use Android is the HTC G1 Dream and it is sold as a "version for developers".

In summary, as well as the Linux platforms mentioned previously there are others such as the Ubuntu Mobile and Embedded project that attempt to be a version of the Ubuntu Linux operating system adapted to work in mobile devices.

Even though many companies participate developing projects for the Linux based operating systems for mobile devices, the market share for these is only significant in Japan and China, used by some popular Motorola devices in those countries, where it directly competes with Symbian. Its market share is approximately 40%, number obtained by adding all mobile Linux based architectures.

Now there is great expectancy with the Google platform, Android. Hope is placed on Android, that as an open standard for mobile platforms it will avoid the fragmentation of operating systems and devices. The question is: Will Google, with its powerful platform, Android, end fragmentation and impose itself on the giants of the mobile devices?

### 3.5.1 Developing for Android

An Android application can be developed using the same tools used for developing in Java. The Android core libraries offer all the necessary tools to build impressive and powerful applications for mobile devices.

The Android SDK supports Windows XP, Windows Vista, Linux and Mac OS and uses Eclipse as development IDE through a plug in, Android Development Tools (ADT), although it is possible to use Apache Ant for Mac or Linux environments.

#### Anatomy of an Android application<sup>5</sup>:

The Android applications are made up of a combination of the following blocks:

- **Activity:**

An "activity" is the most common of the Android application blocks. Usually, an "activity" is a single screen in your application. Each "activity" is implemented as a class that extends the "Activity" class, which displays a UI made up of "Views" and responds to events. Usually an application consists of multiple screens. For example, a messaging application could use a screen to show the list of contacts, a second screen to write the message for the selected contact and other screens to modify settings. Each of these screens is implemented as an "Activity". Browsing between screens is carried out initiating a new "Activity". In some cases, an "Activity" could return a value to the previous "Activity", for example "Activity#1" could allow selecting a photo and this photo would be returned to "Activity", who made the call to "Activity#1".

When a screen is opened, the previous one is set to pause and added to the "History Stack". The user can browse previous screens invoking the screens filed in the "History Stack". Screens can be removed from the "History Stack" when their filing is inappropriate. Android keeps a "History Stack" for each one of the applications that are initiated from "Home Screen".

- **Intent and IntentFilter:**

Android uses a special class called "Intent" to move from one screen to another. An "Intent" describes what the application does. The two most important parts of the "Intent" data structure are the action and the data on

---

<sup>5</sup> Source: <http://celutron.blogspot.com/2007/11/manos-la-obra-anatoma-de-una-aplicacin.html>

which it will be carried out. The typical values for action are "MAIN", "VIEW", "PICK", "EDIT", etc. The data are expressed as an URI. For example, to view a person's contact information it might be necessary to create an "Intent" with the action "VIEW" and the data defined as an URI that represents that person.

A class related to "Intent" is "IntentFilter". While "Intent" is effectively a request to perform something, an "IntentFilter" is a description of what an "Activity" tries to do (or tries to receive). An "Activity" that is capable of displaying a person's contact information could declare how to treat the action "VIEW" when applied to data representing a person. An "Activity" declares its "IntentFilter"s in the "AndroidManifest.xml" file.

Screen to screen browsing is achieved through the resolution of intents. To browse forward, an "Activity" calls "startActivity(myIntent)". Then the system examines all the existing "IntentFilter"s for installed applications and takes the "Activity" whose "IntentFilter"s best match "myIntent". The new "Activity" is informed of the "Intent" and starts up. This "Intent" resolution process takes place at run time when "startActivity" is called providing two key benefits:

- An "Activity" can reuse functions from other components by only placing a request in the form of "Intent".
- An "Activity" can be replaced at any moment by a new "Activity" that has an equivalent "IntentFilter".

- **IntentReceiver:**

An "IntentReceiver" can be used to run an application as a result of an event, for example, if there is an incoming call, if the data network is available or if it is midnight. The "IntentReceiver"s don't display a UI, but they can use the "NotificationManager" to alert the user when something is going on. The "IntentReceiver"s are stored in the "AndroidManifest.xml" file, but they can also be programmed to be stored using "Context.registerReceiver()". To invoke the "IntentReceiver"s the application does not need to be running. If the application is not running, the system can start it when one of its "IntentReceiver"s is triggered. The applications can send their own "Intent"s to other applications through the Context.broadcastIntent().

- **Service:**

A "Service" is an application that is kept active for a long time and does not display a UI. An example is a program that produces mp3 files from a music list, while the user performs other tasks. In this case, the program could initiate a service with "Context.startService()" and in this way, play music without the need of a screen. The system keeps the music reproduction going until it ends. It is possible to contact a "Service" using the "Context.bindService()" method. Once the application is connected to the service, the communication between the application and the service is carried out through the service interface. For the example of the mp3 reproducer, this interface could allow pausing or skipping to a new song.

- **Content Provider:**

Applications can save their data in files, in a "SQLite" database or any other mechanism. However, a "Content Provider" is useful when the application data

must be shared with other applications. A "Content Provider" is a class that implements a standard set of methods for other applications to save and recover the type of data handled by that "Content Provider".

## 4. Conclusion

Although there is no agreement between manufacturers on the definition or name of the mobile devices, it is necessary to differentiate the smartphones with sensitive touch screens from those that do not have one. This division is necessary due to the differences between managing a mobile device used with one hand and generally two customizable buttons and a cursor control (which can be a wheel, a track ball or a five direction button) and manage a device built to use both hands and select actions from any area on the screen, either using the fingers or a special pointing pen.

What is true is that a smartphone has an identifiable operating system and development kits available, whether they be commercial or free. As for the market share, the US market for mobile devices has a very different distribution to that of the rest of the world, with several distinct characteristics:

- In the rest of the world, with market shares higher than 50%, the leader without doubt, is Symbian, while in the US Symbian has scarce participation.
- The main operating system in the US market is RIM (BlackBerry), a system thought for business applications which won its market share thanks to the e-mail technology. This technology is already available in the devices of its main competitors, Windows Mobile 6 and iPhone.
- The US market continued growing in 2008, by almost ten points, while the Japanese, European and Chinese markets retreated.
- There is great expectation with Android, backed by Google, who with only one device on the market (version for developers) obtained sales equivalent to 4% of the US market in one quarter. The Linux community hopes that this system may unify the Linux based mobile market fragments, with scarcely compatible architectures, as its use is free. Motorola (major mobile manufacturer in the US) recently announced that it is leaving Symbian (consortium dominated by Nokia since 2008 Q4) and plans to build Android based mobiles by 2009. It is interesting to note that the Linux based mobiles have a market share of 40% in the Asian markets.
- The operating system for Palm devices, called Access, only present in the US market, was replaced during the current year by Windows Mobile. This was announced by the company after removing from the market its first mobiles with this operating system.

### Development environment costs and the SDKs

While iPhone, RIM and Symbian are free, Microsoft requires a Visual Studio 2005 license to install its SDK.

All the applications that require resources considered "non basic" (multimedia access, GPS) must be signed with certificates, which cost around u\$s200 per year. The iPhone applications are an exception in that they need a certificate to be installed in the test phone and a second certificate to be able to distribute and install them in any iPhone. These certificates are provided by Apple and their cost is around u\$s 99, each one.

Finally, by 2009, Windows Mobile 7 release is expected. According to what Microsoft announced, there will be an intense use of the touch screens and GPS systems and it will

include iPhone and Android characteristics, such as the html rendered directly on screen. Google has already announced its mobile final version global market release for the first semester of 2009, without restrictions for operators.

### General Summary

Device	SDK	OS	Browser	Approx. Cost (Argentina)
<b>BlackBerry</b>	<ul style="list-style-type: none"> <li>• Plugin for VS .Net C# (Free)</li> <li>• Java (Free)</li> </ul>	<ul style="list-style-type: none"> <li>• RIM</li> </ul>	<ul style="list-style-type: none"> <li>• OperaMini</li> <li>• Minuet</li> </ul>	<ul style="list-style-type: none"> <li>• Without Touch:\$1100</li> <li>• With Touch:\$2400</li> </ul>
<b>HTC</b>	<ul style="list-style-type: none"> <li>• Visual Studio .NET (as from U\$S)</li> <li>• <del>Android</del> Android (Free)</li> </ul>	<ul style="list-style-type: none"> <li>• <u>Without touch screen:</u> Windows Mobile SmartPhone</li> <li>• <u>With touch screen:</u> Windows Mobile Professional and Android</li> </ul>	<ul style="list-style-type: none"> <li>• iExplorer</li> <li>• OperaMini</li> <li>• Android</li> </ul>	<ul style="list-style-type: none"> <li>• Without Touch:\$900</li> <li>• With Touch:\$1700</li> </ul>
<b>iPhone</b>	<ul style="list-style-type: none"> <li>• Cocoa Touch (Free)</li> </ul>	<ul style="list-style-type: none"> <li>• MAC OSX</li> </ul>	<ul style="list-style-type: none"> <li>• SAFARI</li> </ul>	<ul style="list-style-type: none"> <li>• With Touch:\$2000</li> </ul>