



Aalborg Universitet

AALBORG UNIVERSITY
DENMARK

Mobile Platforms

-An analysis of Mobile Operating Systems and Software development platforms

Hammershøj, Allan; Sapuppo, Antonio; Tadayoni, Reza

Publication date:
2009

Document Version
Publisher's PDF, also known as Version of record

[Link to publication from Aalborg University](#)

Citation for published version (APA):

Hammershøj, A., Sapuppo, A., & Tadayoni, R. (2009). *Mobile Platforms: -An analysis of Mobile Operating Systems and Software development platforms*. Paper presented at CMI International Conference on Social Networking and Communities, Ballerup, Denmark.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- ? Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- ? You may not further distribute the material or use it for any profit-making activity or commercial gain
- ? You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us at vbn@aub.aau.dk providing details, and we will remove access to the work immediately and investigate your claim.

Mobile Platforms

-An analysis of Mobile Operating Systems and Software development platforms

Allan Hammershøj, Antonio Sapuppo and Reza Tadayoni
Center for Communication, Media and Information technologies (CMI)
Aalborg University
{allan, antonio, reza}@cmi.aau.dk

The paper is developed for presentation at:

CMI international conference on social networking and communities

25-26 November 2009

Copenhagen, Denmark

Abstract:

Mobile Social Networking is becoming a reality driven by the introduction and further development of smartphones. Mobile communication has been dominated by vertically integrated service provision in a 'operator centric model', which has been highly bound to voice and SMS services and organized in a monopolistic competition between few Mobile network operators, MVNOs and SPs. In the recent years we are witnessing a radical change driven by introduction and further development of smartphones, where we see emergence of new business models, including 'device centric models', where the user can get access to new applications and services by connecting to the device manufactures' app stores and the like. One of the main drivers of this change is the advanced capabilities of the smartphones enabling the mobile devices to reap the advantages of the convergence process and bring the advanced Internet applications and services like social networking to the mobile devices. However, the device market is dominated by a number of different technological platforms, including different Operating Systems (OS) and 'software development platforms', resulting in a variety of different competing solutions on the market driven by different actors. The aim of this paper is to give a comparative analysis of these technological platforms and identify their strengths and weakness for being *the* platform of the future.

Keywords:

Convergence, Mobile OS, Mobile software development platforms, Mobile Internet, smartphones

Status:

Work in progress

1. Introduction¹

WEB 2.0 services and applications are increasingly going mobile. One example is the Mobile social Networking, which is the step forward in the development bringing the social networking applications and services to the mobile devices and adopting them to the specificities of mobile devices, being personal and aware of the context the user is in. These two characteristics are seen as vital in development of advanced mobile services and applications, including the social networking applications. See amongst others (FEIJOO et al., 2009).

Huge penetration of mobile devices, in particular smartphones, and the development of mobile broadband are important factors in the development of Mobile Social Networking applications and services. At the end of 2008 close to half of the world's population had a mobile phone and about 54% of telecom revenue came from mobile services (IDATE 2009). The smartphone penetration is still low at approximately 8% which means that these top-end devices remain niche (KHANA 2009), but is developing fast. The bandwidth capacity in mobile is increasing fast in the advanced markets. According to (IDATE 2009) the number of 3G customers in EU-27 in 2012 will become about 4 times more than 2008, going from 111 million in 2008 to 412 million, indicating massive growth in the bandwidth capacity in the mobile network. In addition to this, the smartphones will also have access to other high bandwidth networks like WiFi.

The mobile industries have great expectations in mobile social computing. In (FEIJOO et al., 2009) the authors, based on own calculation and a number of analytic reports, forecast that the world revenue will grow from about 1 billion € in 2008 to 7-8 billion € in 2013 hereof the mobile social computing part will be third, after music and gaming.

In the recent years we are witnessing a radical change driven by introduction and further development of smartphones, where we see emergence of new business models, including 'device centric models', where the user can get access to new applications and services by connecting to the device manufactures' app stores and the like, see amongst others (BALLON 2009). One of the main drivers of this change is the advanced capabilities of the smartphones enabling the mobile devices to reap the advantages of the convergence process and bring the advanced Internet applications and services like social networking to the mobile devices.

However, the device market is dominated by a number of different technological platforms, including different Operating Systems (OS) and 'software development platforms', resulting in a variety of different competing solutions on the market driven by different actors. This fragmentation of technological platforms and standards are seen as a barrier for development of content and services, which locks the users to specific technologies or puts an immense load to the content and service providers to adopt their content /services to all these platforms.

The aim of this paper is to give a comparative analysis of these technological platforms and identify their strengths and weakness for being *the* platform of the future. The analysis in the paper will be based on following parameters:

¹ This paper is developed as a part of the CAMMP project (<http://www.cammp.dk>) funded by the Danish Advanced Technology Foundation (Højteknologifonden)

- **Openness:** The degree of open source is the OS. It also takes into account the accessibility through the available developer tools to the terminal hardware.
- **Look and feel:** The overall impression of the graphical user interface (classic, modern or trendsetting), the general user experience and how the OS launches and handles the vast amount of applications.
- **Web integration:** The implementation of the OS towards being an online community terminal client and integration towards cloud computing in general.
- **Industry and community support:** The two main drivers of software and hardware today. The companies are on one side putting a lot of effort and money in promotion and development, and the user communities on the other side supporting the OS by developing applications for it.
- **Future perspectives:** The strategy evaluations and predictions done by other analysts on how the OS is going to perform in the growing smartphone market in the future.

Furthermore, the **business model** of the major platforms and the major deployed **software development platforms** are analyzed and their effects on the mobile platforms are discussed and evaluated.

The paper is structured as follows: Following this introduction in chapter 2 the market and business models for delivery of advanced mobile services and application are discussed and case study of major implementations are given. Later in Chapter 5 the major Operating Systems (OS) are discussed and compared to each other. After this in Chapter 4 an analysis of different development platforms is given. Finally in chapter 5 the conclusions, including the main drivers of the development are discussed. Chapter 6 contains references.

2. Market and Business model

Social networking is one of the main applications discussed and analyzed in the literature under the umbrella of WEB 2.0 technologies and applications, being a set of technologies and applications that enable interaction and user participation rather than static viewing and consumption which characterizes the WEB 1.0 paradigm (O'REILLY 2005), (ANDERSON 2007), (HOEGG, et al. 2008). This distinction has been challenged by some of the pioneer in WEB development arguing that the original WEB was meant to be interactive and the reason for the development of 'static WEB' in the beginning was that: '...during a series of ports to other machines from the original development computer, the ability to edit through the web client was not included in order to speed up the adoption process...' (BERNERS 1999). Nevertheless in this paper we stick to the above mentioned 'umbrella term' WEB 2.0, being a set of technologies and applications that enable interaction and user participation.

The social networking and user participation in production has been an important contribution to value proposition in the WEB 2.0 applications and has enabled a number of new actors to take part in the development and innovation process. In (POUWELSE et al., 2008) it is argued that the evolution of peer production shows that value creation and innovation is increasingly moving away from telecommunication and content industries towards "the edges" . With respect to the extent of newcomers getting involved in the market, according to (FEIJOO et al., 2008) already in 2007, almost half of the mobile content and application start ups were aimed at social networking area.

Generally two modes of delivery of advanced mobile services and applications can be identified: The 'app store model' and the 'browser based' model. In the app store model the users will access the app store and acquire the application, install it on their smartphone and use it. An example for the app store model is Apple / iPhone. In the browser model the users access the service through a browser on the mobile device. The service provision is then organized in different ways discussed in the following.

Mobile communication has been dominated by vertically integrated service provision in a 'operator centric model', which has been highly bound to voice and SMS services and organized in a monopolistic competition between few Mobile network operators, MVNOs and SPs. In the recent years we are witnessing a number of different business models, in particular when it comes to provision of advanced application and services.

In (FEIJOO et al., 2009) the business model of mobile industry divided in two main categories: 1) the "walled Garden" model, which is characterized by operators' prominent position, controlling things from networks and applications to services and content. 2) The "dumb pipe" model, where the mobile operator is envisaged as provider of connectivity. Here the revenue of mobile social computation applications would accrue to providers, enablers and brokers, in a scheme similar to that of broadband access to the Internet.

In a recent paper titled 'The platformisation of the European mobile industry' (BALLON 2009), Pieter Ballon goes a step further and suggests four different business models (platforms) with different gate keepers roles: 1) Telco centric model, where the telecom carrier acts as portal provider, services aggregator, network operator and portal provider, in this model the user accesses services through a portal. Example: Vodafone live. 2) Device centric model, where the main service platform is incorporated in, or tied together with, the mobile device. Example: Apple iPhone. 3) Aggregator centric model, where the actor role of portal provider is taken over by service aggregator independent of mobile operator. Example: Facebook, 4) Service centric model, which is characterized in the paper as merely a 'theoretical model', where the user connects via the mobile network operator and selects services on a case by case basis. Example: Google's Open Social initiative.

The device centric model is interesting as it seems that following the success of iPhone, all different OS vendors try to come with a solution. And it seems like the success we experienced with iMode bringing advanced services and Internet to mobile devices is coming to Europe through this model. On the other hand we are witnessing a fragmented market with a number of different solutions (see Figure 1) that makes development of services and bringing them to mass market a challenging issue.

2012 Preliminary Forecast – 522 Million Units					As of October 2009
Platform	Unit Sales(M)	4Q/12 Share	1Q/09 Share	Difference	
Symbian	203.58	39.0%	49.3%	-10.3%	
Android	75.69	14.5%	1.6%	+12.9%	
iPhone OS	71.51	13.7%	10.8%	+2.9%	
Windows Mobile	66.82	12.8%	10.3%	+2.5%	
RIM OS	65.25	12.5%	19.9%	-7.4%	
Linux	28.19	5.4%	7.0%	-1.6%	
webOS	10.96	2.1%	0%	+2.1%	

Figure 1: Platform market share²

In the following short description of major platforms are given mainly seen from the business model perspective. More information about these and other platforms like Windows media, Sony Ericsson etc., can be found in amongst others (KHANA 2009). Later in this paper we give a technical analysis of major operating systems and development platforms to identify the strengths and weaknesses of these platforms.

2.1 Apple/iPhone

The following time line shows a rough overview of the history of Apple/iPhone:

- 1976: Apple Computer Inc. was founded
- 2001: Apple introduced iPod on the market
- 2007: Apple Computer Inc. changed name to Apple Inc.
- 2007: Apple introduced iPhone and iPod Touch
- 2008: Apple launched Apple app store

The change of the name and removing Computer from the name in 2007 is a signal from Apple indicating a strategic change from being mainly in the computer industry to taking part in the broader media convergence industry. iPhone is Apple's materialization of the convergence between media, internet and telecoms. With its large screen, good hardware/software capabilities like 2G/3G and WiFi, network connectivities, GPS, modified Safari browser, camera, motion sensors, touch display , and the user friendly interface to the Internet, media and telephony services iPhone has been a huge success and a trend leader in the smartphone market.

When iPhone was launched it was radically different from the way smartphones were designed. Available smartphones at that time were more like extension of a PC and targeted towards business people, where the iPhone was designed for the masses as well. It has been trendsetter for interfacing with a user providing multi touch support and access to a vast amount of applications and software using iTunes Store, App Store (see below) and MobileMe. The latter with focus on synchronising the desktop with iPhone like calendars, photos and files in general. It comes with a YouTube client preinstalled and the newest version support HD video playback up to 720p.

² <http://www.mobilecrunch.com/2009/10/08/analyst-android-to-become-2-mobile-os-by-2012/>

The revenue of iPhone comes primarily from sale of the terminals. Here we have a major difference between the iPhone and other platforms like Vodafone live and iMode. The primary revenue source for Vodafone and NTT DOCOMO are the generated traffic as none of them are handset companies. In the beginning Apple also got shares of the generated traffic, this was due to Apples marketing strategy, where they developed partnership programs with specific operators in different markets to exclusively distribute their handsets and at the same time got share in the generated traffic. For example, according to guardian in September 2007 'O2 would return to Apple as much as 40% of any revenues it makes from customers' use of the device'³. However Apple changed this strategy later to enable faster diffusion of the devices on the market⁴

Another specificity of iPhone, that makes it merely a platform rather than a sophisticated handset with a specific marketing strategy, is the launch of 'App Store' in 2008. Here Apple has created a market place where the application providers can meet the costumers and sell their applications directly to the costumers. The model is then that Apple gets a share of the revenue, with a 70/30 split, where 70% of the revenue goes to the developer and 30% to Apple (BURKHARD 2009). On app store it is possible also to get free applications. With regards to these applications Apple does not get any revenue, however to be part of the program all developers must pay 99 US\$ per year which also covers the access to Apple's development kits etc. (BURKHARD 2009).

With regards to the applications, the modified Safari browser enables access to regular web pages, which may be seen as an important advantage as it is not necessary to optimize the web sites for use on iPhone. However some service providers do adopt their websites to iPhone and the iPhone version of a website can either be accessed by a URL or it delivers it automatically upon detection of the iPhone browser by the web server (BURKHARD 2009).

2.2 Google/ Android

Google acquired the small mobile software developer company, Android, Inc. in July 2005. This was obviously a move from Google to extend their successful business on the Internet also to include the mobile market. The aim has been to develop a common open source operating system that any mobile device can run on and to make developing applications for mobile phones more general and hardware independent.

The development of Android is taking place in the framework of 'The Open Handset Alliance' (OHA) with a number of major technology and mobile companies⁵. 'HTC Dream' is the only Android-enabled mobile phone on the market.

Currently, the Android Market charges 25 USD to be a registered developer. As well, a developer can purchase and Android Developer's Phone for almost 400 USD. Royalties could potentially be charged for non-free applications on the Android Market as well (KILGO 2008).

In table 1 a number of differences between Android and iPhone are listed. Both platforms offer a way for developers to publish their applications either for free or for a price. Also users can visit and download

³ <http://www.guardian.co.uk/technology/2007/sep/17/mobilephones.apple>

⁴ <http://news.cnet.com/apple-unveils-iphone-2/>

⁵ For more information about Open Handset Alliance refer to their web site: <http://www.openhandsetalliance.com>

applications from the Apples app store or Android marketplace. We are witnessing a number of applications developed for iPhone becoming available on Android and vice versa.

In (HEATLEY & HOWELL 2009) different characteristics of Apple/iPhone and Google/Android are discussed and compared. This is summarized in Table 1 which outlines some important attributes of these two services.

	<i>Apple/iPhone</i>	<i>Google/Android</i>
Brand Values	High quality	Universality
	Clean property rights	Free – take what you get
	Trustworthy Takes responsibility	Open access – get the good and the bad
	Mediated	Peer-to-peer
	Consistent Highly useable	Less consistent, at some cost of usability
Third party Access	Regulated and quality controlled	Unrestricted
Target Consumer Attributes	Specialists	Generalists
	Dislike of advertising	Open to advertising
	Time poor	Time rich
	Price inelastic	Price elastic
Customer relationships	Direct	Anonymous
Core Content	Paid music and videos	Search and mapping
Payment	Cash	Eyeballs (viewing advertising)

Table 1: Selected attributes of the Apple and Google bundles (HEATLEY & HOWELL 2009)

As mentioned earlier an important difference between Android and iPhone platforms is that Android makes no attempt at being specific to a specific hardware and offers drivers for many of the common types of hardware on the mobile phone.

2.3 Other platforms

In the following a short description of other important platforms is given

2.3.1 Nokia / OVI

Nokia has also created its own ‘app store’ / ‘market place’, which is called **OVI store**. OVI store can be accessed by Nokia’s Symbian devices. It is still relatively new, which can be seen on the limited amount of applications available in the store compared to iPhone and Android. The applications are not as creative and advanced in functionality and graphics. However, OVI is a major strategic move from Nokia and is an important to follow in the future.

2.3.2 Blackberry

The Blackberry also created an application store called **App World**. App world has some applications but far from the vast amount in the application store for the iPhone. These applications are more traditional like getting world weather, access to existing web services with less focus on the graphical part.

2.3.3 Windows Mobile

Alongside with the introduction of the new update of Windows Mobile (version 6.5) in October 2009, windows launched access to a new application store from Microsoft called **Windows Marketplace**. The

amount of applications available is limited and the functionality is not creative and fancy like iPhone and Android. The Microsoft application store is not the only one for the Windows Mobile. Due to the many years on the market existing web sites contains a lot of applications available to the device. Most of them lack, however, the creative layout and innovative steps known from the iPhone.

3. Mobile Operating Systems (OS)

This chapter gives an analysis of the mainstream operating systems for mobile terminals. Not all of the available mobile operating systems on the market have been analysed. Only the most active and company driven variants have been selected based on market analysts' factual numbers (and predictions (Gartner, 2009), (DigiTimes, 2009)). One common factor is that all of the operating systems are targeted for smartphones.

3.1 iPhone

The iPhone (and iPod Touch) OS is developed by Apple. It is currently in version 3.1 and is based on a variant of the same Darwin operating system core that is found in Mac OS X. the "Core Animation" software component from Mac OS X v10.5 Leopard is also included. The OS is capable of supporting bundled and future applications from Apple, as well as from third-party developers.

The OS does not support multi-tasking making the switch between applications slow. In particular, using web services and games with login information suffer from this. Applications development for the iPhone OS is mainly done using objective C, but C/C++ development is also possible. Effort has been put in supporting Web Runtime (WRT) widgets as well. These widgets are written in JavaScript, CSS and HTML that are supported on most smartphones and web browsers in general. This is due to the fact that most available browsers (except Firefox and Internet Explorer) are based on WebKit. WebKit is an open source web browser engine used in Apple's Safari, KDE's Konqueror, Nokia's S60 browser, Google's chrome and many more. It has also been ported to Qt (cf. sect. 4.3). Composing applications using WRT widgets, is popular on many mobile OS. webOS and LiMo being front-runners. The support for WRT do not include the opportunities to work with Flash, as the iPhone OS does not officially support Flash in any version.

Development of the iPhone OS is controlled by Apple in all aspects. However, the short learning time to start building applications to it, that even takes advantage of the hardware possibilities, together with the large community around it are the main drivers. However to keep the position of the iPhone, Apple will have to continue development and innovations, as an iPhone is an expensive product in a market with many relatively even competitors. As of today it seems iPhone to be the most popular smartphone available (Helmreich & Doriot, 2009).

3.2 Android

The Android OS is a result of the Open Handset Alliance (OHA) with Google as one of the very active partners in the implementation. Other notable partners in OHA being handset manufacturers are HTC, LG, Samsung, Sony Ericsson, Motorola and NTT Docomo - not Nokia, Palm and Apple. Android is designed with the intention of working fast on even small devices (with the ARM hardware architecture) and be as open source as possible.

It has recently been released in version 2.0. Android OS isn't made in Java, but the application development for Android that is in Java. However, C/C++ and ARM Assembly can also be used when using a Native Development Kit. Android is based on a Linux kernel with the user space and the JVM for Android (Dalvik) being written in C. The Java implementation is based on a custom profile with a lot of own developed functionality especially regarding the graphical components and processing.

The OS is fully open source; however Android applications created by Google to access existing Google web services are not open source and not allowed to be distributed without permission from Google. The operating system is in rapid development and the new version 2.0 has improved on several functionalities: now better supporting multitasks, add of multi-touch support and double tap zoom. Multi-tasking functionality is an important and powerful feature of Android. Launching applications into the memory is very slow (several seconds), but are quickly fetched from the cache again. It is assumed that this will improve in later versions along with an official support of using SD for storing applications (9to5mac, 2009). The graphical look of the user interface is modern and takes full advantage of the touch screen.

The OS is implemented with the vision of being always on and most processing will happen online making this OS a suitable choice for developing cloud computing and social communities applications. This is Google vision in general and why the upcoming (November 2009) Google Chrome OS is based mainly on the concept of cloud computing. Android does support the full Adobe Flash and the newest version even support Adobe Flash 10.

Several hardware manufacturers are now announcing Android enabled terminals (smartphones and netbooks), even companies that normally are known for ordinary PC hardware (like Dell and Acer).

3.3 Symbian

The Symbian OS is the most popular mobile OS in the world so far. However, the introduction of many much more modern competing OS on smartphones that cleverly have used touch screen, accelerometers, implemented application stores, support for WRT, easy access to the hardware for application developers and much more, has really challenged the future strategy of Symbian OS.

Nokia has been the main actor using Symbian for their OS implementation called the S-series. Other handset manufacturers are using Symbian as well, but the future of Symbian has heavily been influenced by the decisions of Nokia. The most advanced of the S-series OS was the S60 that from version 5 supported touch screen as a reaction to the iPhone success. None of the S-series OSs from Nokia have been open source so far⁶.

Development of open platforms designed by different telecom stakeholders in collaborations like OHA and the fact that iPhone was growing in the smartphone market, could have been one of many reasons that Nokia decided to buy Symbian and form the Symbian foundation. Symbian foundation, besides Nokia themselves, currently consists of notable terminal and chip manufacturers (Samsung, LG, Sony Ericsson, etc.), software developers and telecom operators - many also part of OHA. The decision to form the Symbian foundation has to seen together with the decision earlier same year to acquire the Norwegian

⁶ The alternative MAEMO OS originally intended for Internet tablets built around a Linux kernel was the first step towards an open source OS from Nokia

Trolltech inventors of Qt (cf. next section). Both are major investments and the next version of Symbian will become open source. A drastically change in strategy.

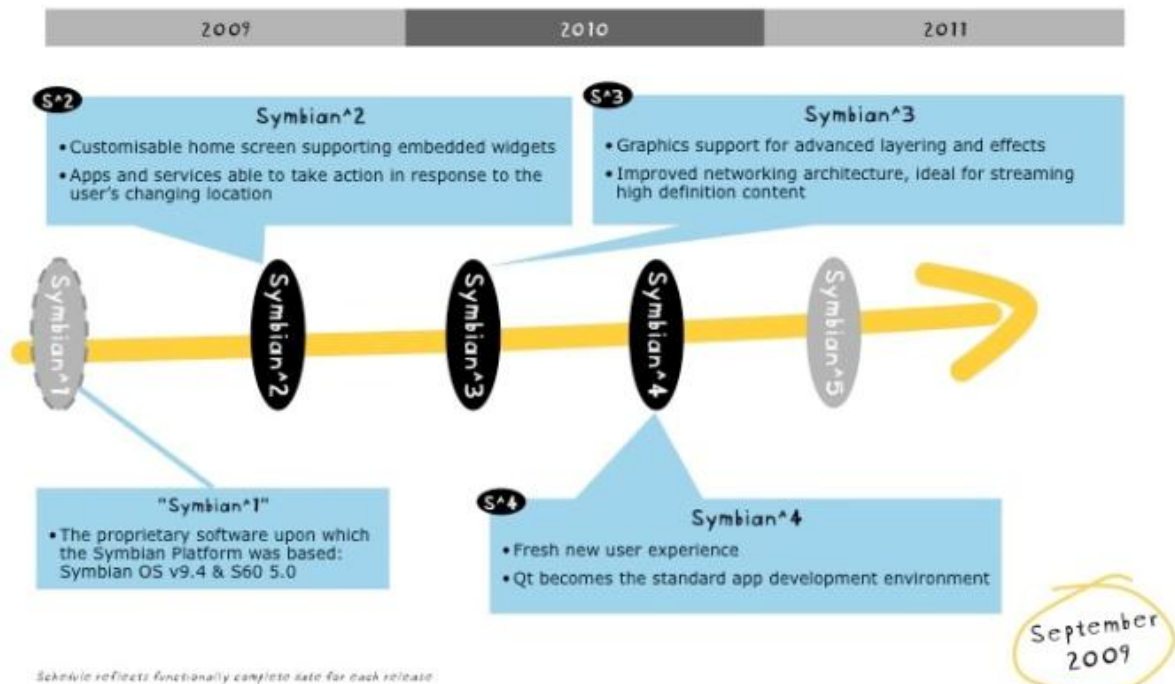


Figure 0: Roadmap for current and development of future versions of Symbian (Symbian, 2009).

The S60 version 5 was designed on top of the Symbian v9.4. This implementation has been renamed to Symbian^1. Next version will be the first open source version of Symbian called Symbian^2 aiming at implementing WRT, Symbian^3 will focus on the graphical experience and better support for streaming video and from Symbian^4 a complete new user interface with all graphical components and standard application development based on Qt. This is illustrated in figure 2.

Symbian originates from the Psion's graphical OS called EPOC primarily designed for PDA back in the 1980s. It was, and is still, a great multi-tasking OS and the newest release version called Symbian^1 provides a developer the opportunity to implement applications in Qt, Python Mobile, J2ME, Flash Lite (not standard Flash), Ruby, .NET, WRT Widgets, Symbian C++ and Standard C/C++. It seems like Symbian in a future version will furthermore support BONDI (cf. sect. 3.8) (Bloor, 2009) along with support for multi-touch. Also steps towards bringing Symbian closer to the hardware has been taken with the forthcoming Symbian Hardware Abstraction Interface (SHAI).

For the analysis of the OS, the devices running Symbian^1 has been used for testing. Running the different applications on the device is not working specifically fast. Worst case launching applications takes several seconds. Accessing the OVI application store is very slow. However, switching between applications as multi-tasking is working very efficiently. Synchronising the content of folders and calendar is possible using the internet, but in general it feels a bit traditional like with the Blackberry OS. On the contrary to, e.g., Android it does not feel really prepared for taking advantage of cloud computing and be empowered by online social communities. The look of the desktop and menus in general are old-fashioned and lacks the innovative control like webOS and especially iPhone.

All in all Symbian¹ is an OS that seems being a generation behind competing mobile OS. However, looking at the roadmap for the further development of Symbian and other supporting open source initiatives done by the Symbian foundation with Nokia in the lead, it seems that Symbian will still be a strong competitor on the smartphone market in the future.

It has recently been indicated by developers at Nokia that they will stop using Symbian OS in their high-end N-series smartphones from 2012 (Nokia, 2009) to focus on MAEMO instead. Other rumours have indicated that Samsung will start phasing out Symbian from 2010 to focus on Windows Mobile, Android and their own developed Linux based mobile OS called Bada⁷. Both statements have later been officially denied by Nokia (Nokia deny, 2009) and Samsung (Samsung deny, 2009) respectively.

3.4 Blackberry

The Blackberry OS and development platform is developed by the Canadian company Research-In-Motion (RIM), and was released in its latest version 5 in October 2009. The OS is providing a platform for doing application development supporting solely J2ME the profile CLDC based on the profile MIDP. The Blackberry Java Virtual Machine (JVM) is based on Sun's implementation of the J2ME being written partly in C, C++ and assembler. It is a native implementation located in the actual firmware of the device, making it very hard to hack or in any way alter. The two greatest advantages of this are that: 1) the OS doesn't have to be compiled to the CPU type of the device, and at the same time 2) it provides a hardware abstraction layer to other hardware functionalities of the device like button control, sound, radio communication etc. On paper this gives a better device performance eliminating many bottlenecks in hardware access.

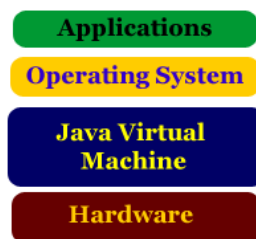


Figure 3: A conceptual and generalised model for a Java based operating system. Applications are executed in the OS using a JVM with a specified profile and configuration.

Figure 3 illustrates how the Blackberry platform is placed in a data layer model. This is however not entirely the truth. The actual Blackberry OS is written native along with the JVM and all Java libraries as well as the internal implementation of the java libraries is made native. However for simplicity the OS is placed on top of the JVM as all development and the preinstalled applications supports J2ME.

The operating system is implemented supporting true multi-tasking without noticeable performance lack and due to the implementation of the OS; the applications load fast and perform really well on slow devices. The general layout and browsing through menus feels classic but it works smoothly. Multi-touch is supported in the newer versions of the OS. Blackberry OS does not support any Flash versions and the use of the internet seems to be intended for surfing, e-mail and traditional client server calendar synchronisation.

⁷ Bada has not been considered in this article as the OS has not yet been officially presented. This will happen in December 2009 (www.bada.com/category/blog/ceremony/).

The Blackberry was originally adopted by business people due to especially the clever e-mail handling and it feels like this is still the case. RIM solely controls the development for the OS and it will probably be around for years to come. But especially Android, iPhone and the future Windows Mobile 7 enabled smartphone can be hard competitors.

3.5 Windows Mobile

Windows Mobile OS is a proprietary and not open source OS originally created to be a mobile version of Windows with a user interface compliant to the current Windows version. It is based on the Windows CE v5.2 kernel, which most hardware specific components are offered as open source. Windows CE is a minimalistic real-time multi-tasking OS that can run in less than a megabyte of memory.

Windows Mobile was updated to version 6.5 in October 2009 and the analysis is based on this update. With this update a new user interface inspired by trends from smartphones user interfaces. The update also included access to the new application store from Microsoft called Windows Marketplace. The amount of applications available is limited and the functionality is not creative and fancy like iPhone and Android. The Microsoft application store is not the only one for the Windows Mobile. Due to the many years on the market existing web sites contains a lot of applications available to the device. Most of them lack the creative layout and innovative steps known from the iPhone.

A number of programming languages are supported for Windows Mobile and CE development. This includes J2ME, Qt, Visual Basic, .NET, Visual C++ and a lot more. Together with support for WRT widgets (JavaScript, CSS and HTML) and Flash Lite as well Windows Mobile must be considered a very versatile platform for application developers.

The feeling of the OS depends solely on the device selected for implementation having application launch times ranging from a few to several seconds. On the better devices the OS feels solid, stable and quick to switch between applications in when multi-tasking. The synchronisation with a PC or other Microsoft software (like Exchange servers) works smoothly and the OS comes with a lot of business related applications preinstalled. All in all this makes the OS a good choice when being a business customer needing a PIM. The user interface from v6.1 has been updated to v6.5, but it still looks and feels generations behind webOS, Android and iPhone (Helmreich & Doriot, 2009). It has been losing terrain in the growing smartphone market (Ars Technica, 2009), and it sounds like the version 6.5 will not help this tendency. Gizmodo has made two different analysis of the OS and they have rated it very low (Gizmodo Windows Mobile, 2009), (Gizmodo HTC, 2009). There are indications that even Microsoft finds their OS outdated to some extend (Ballmer, 2009).

Like the Symbian Foundation with the Symbian OS, Microsoft is putting all of its effort into regaining a position at least as the preferred PIM for companies with the forthcoming Windows 7 (rumoured spring 2010).

Looking at the Microsoft multimedia player Zune's user interface (Zune 2009) shows a much more modern approach to a user interface. Holding this together with the plans of implementing parts of it in Windows Mobile 7 (Zune Windows Mobile, 2009), compare it to the few available previews (Windows Mobile 7 preview, 2009) and the promotional trailer from Microsoft (Windows Mobile 7 trailer, 2009), the new OS looks to be really reengineered in many aspects.

3.6 webOS

The webOS OS is the classic PDA maker Palm's newest version. It is very different from older versions, as this is solely based on applications using WRT with HTML, JavaScript and CSS. No versions of Flash are supported. Currently there only exists two devices with webOS and the newest is from mid November 2009.

webOS is a fully open source OS based on a Linux 2.6.24 kernel with proper patches and hardware driver support together with a WebKit implementation (cf. sect. iPhone) that is the engine for the applications. In webOS terminology applications are composed of one or more distinct scenes, rendered on a stage in a card. A stage can be compared to a window or tab in a browser, and each scene as different web pages. Each scene is composed of underlying JavaScript code that provides the functionality of the scene. Within a scene, the application presents graphical widgets to the user that the user can interact with. This conforms to the W3 Document Object Model (DOM) (W3, 2009) providing access to all elements on a web page that can be dynamically updated or manipulated. A webOS application is either launched by the user or by another application. webOS supports multi-tasking, but applications can be designed to run only in the background, and therefore not present a card to user at all. These background applications interact with the user mainly by displaying alerts and notifications.

The fact that webOS is a WRT OS, the need of an application store seems mandatory. The available application store is called Palm App Catalog and currently there are but a few amount of applications available, but the store has only been open for contributions since August 2009 (App Catalog, 2009). The applications look and feel modern and professional, but in general they launch very slowly (several seconds). Whenever launched the multi-tasking back and forth between the applications works smoothly.

The OS seems to be competing with the Windows Mobile OS and Blackberry OS for the business smartphone segment as being a true personal information manager (PIM). Competing with Android, Symbian or iPhone for the masses do not seem plausible.

3.7 Other important Operating Systems

3.7.1 MAEMO

MAEMO is an open source OS created by Nokia. MAEMO was Nokia's first open source initiative; however, as mentioned earlier, Symbian is announced to become open source as well. It has recently been released in version 5 and all versions are built on a Linux kernel. The authors of this paper have in debt experience with MAEMO through the work on the 6th EU framework program project called MAGNET Beyond. The former versions of MAEMO were not meant for smartphones, but merely for a touch screen internet tablet PC with WiFi and Bluetooth radio communication only. The newest version supports 3G radio communication technologies as well making it possible for a device with the MAEMO OS to be always connected. It support multi-tasking and the upcoming version MAEMO 6 will support multi-touch as well. The older devices running older version of MAEMO will not be compatible with the new version.

The MAEMO OS does not support access to the OVI store for uploading and downloading applications like on the Symbian OS by Nokia. Applications for MAEMO are available through MAEMO select instead. Mozilla has used the OS for developing their first mobile browser called Fennec (using Mozilla Gecko render engine). This along with the facts that MAEMO supports developers to create application in Flash (through

the built-in browser MicroB) along with Qt (with WebKit support), Python (for MAEMO) and C/C++ native language as well ,makes the OS an very promising competitor to the smartphone market.

3.7.2 LiMo

The LiMo OS is created by the Mobile Linux handset trade group called LiMo Foundation. The OS is based on a Linux kernel. The LiMo foundation consists of many telecom stakeholders from various handset manufacturers, telecom operators and software developers. The OS is now in the second version called R2. This version has as the first industry handset support for the Open Mobile Terminal Platform (OMTP) BONDI specification, which is a standard set of secure APIs for mobile browsers or WRT. It provides a consistent and secure Web services interface that can be used by developers across multiple device platforms. BONDI can be said to expose key handset features to 3rd party developers creating mobile applications, while at the same time not compromising the security. BONDI-compliant WRT co-exist with other runtime environments in a device, such as the native operating system, JVM etc.

Key technologies supported by the R2 platform is said to include location-based services, multimedia, personal information management and especially improved security applications. The main application development is intended to be created using web technologies CSS, JavaScript, HTML etc. like webOS. However it supports running applications in a JVM along with native code.

Looking at the lack of openness of the OS (even though it is Linux) though with great support from industry, it is assumed that LiMo will have a chance to get its share in the smartphone market even though the much more open Android and MAEMO, both based on Linux kernels as well, are hard competitors. This is also indicated by the Gartner analysis (Gartner, 2009).

4. Software development platforms

In the previous 2 chapters (chapter 2 & 3) we analyzed mobile platforms seen from market/business and technological perspectives. The latter focused on the OS of the platform. In the following an analysis of the deployed software development platforms/tools is given. We focus on three major platforms: Java 2 Micro Edition (J2ME), Python Mobile and Qt.

4.1 Java 2 Micro Edition

Java 2 Micro Edition (J2ME) is the newest and smallest addition to the Java family. The other members of the Java family are the Java 2 Standard Edition (J2SE) and the Java 2 Enterprise Edition (J2EE). The former is intended for conventional desktop applications development, while the latter one is specifically intended for building distributed applications with emphasis on the server side development and web applications. *J2ME* is intended to build applications running on mobiles and other embedded devices (Mobile Programming, 2007). Figure 4 presents the appropriate Java editions for different kind of devices.

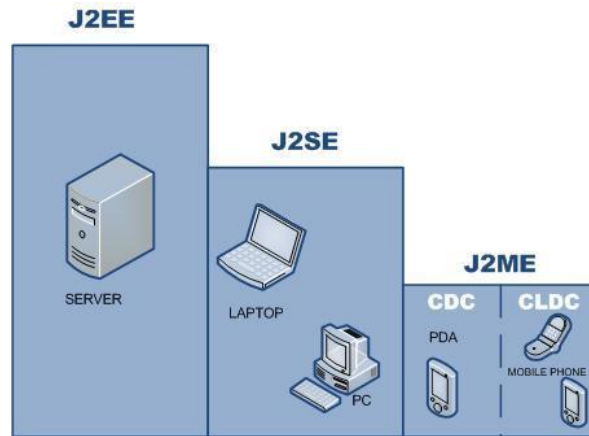


Figure 4 – Java family

In *J2ME* two different configurations exist:

1. Connected Limited Device Configuration (CLDC): intended for more limited devices like mobile phones;
2. Connected Device Configuration (CDC): intended for devices with more memory and faster processors.

Java ME provides a rich set of API for mobile developers as shown in Figure 5.

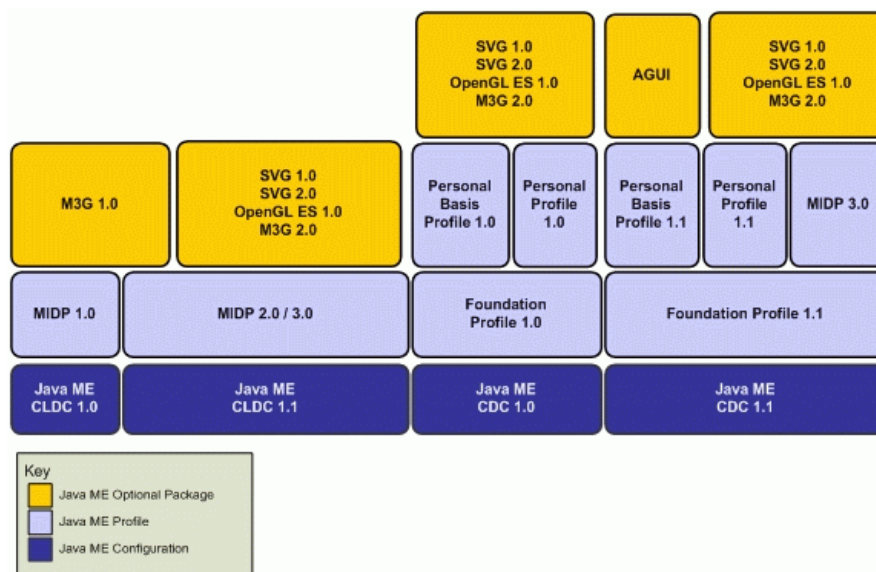


Figure 5 - Java ME API (J2ME, 2007)

As it is seen in Figure 5 the set of API is divided in different layers. In this paper we will have a closer look to the CLDC configuration since it is the one used for smartphones.

While the CLDC contains a subset of the Java-class libraries which give to the developers a solid java platform for creating application, the profiles on top of the configurations includes features such as an enhanced user interface, multimedia and game functionality, greater connectivity, over-the-air (OTA)

provisioning, mechanism to persistently store data and later retrieve it, and end-to-end security (J2ME, 2007).

An optional package is a set of technology-specific APIs that extends the functionality of a Java application environment. CLDC supports a number of optional packages that allow product designers to balance the functionality needs of a design against its resource constraints (CLDC, 2009). Applications written for this profile are called MIDlets.

One of the important benefits of Java 2 Micro Edition is the portability. "Write once, run everywhere" is the slogan of Java applications. Ideally, this means Java applications can be programmed, compiled into a standard bytecode and be expected to run on any device equipped with a JVM.

The next generation of mobile applications using the new Mobile Information Device Profile (MIDP) 3.0 will do the following (MIDP3.0, 2009):

- Run on more capable hardware, with processors approaching 1 GHz in speed
- Run on faster and more ubiquitous mobile data networks, with transfer rates approaching 10 Mb per second
- Multiple radios, including Bluetooth, Wi-Fi, 3G/4G networks, global positioning system (GPS), digital video broadcasting (DVB), and near-field communication (NFC)
- Use various sensors, such as an accelerometer or compass
- Provide rich user interfaces, supporting vector graphics and touch screens

MIDP 3.0 also supports the MIDlet concurrency (where more than one MIDlet can run at the same time, which is a handy feature) and Inter-MIDlet Communication (where two concurrently MIDlets can communicate with each other directly). MIDlets will be able to run in background with no UI and it will be enable the auto-lunched (MIDP3.0, 2009).

Additionally, a better support for devices with larger and secondary display(s) will be provided and it will be enable IPv6, secure RMS stores and Specify standard ways for doing MIDlet provisioning through other means, for example, OMA (SyncML) DM/DS, Bluetooth, removable media, MMS, and JSR 232 (MIDP3.0, 2009).

A major advantage of J2ME is the cross platform issue phrased in the slogan "Write once, run everywhere". This has been implemented by adding a new layer (JVM) that has been blamed for making application runtime slower than with native implemented applications. To defend Java one can argue that this is the price to be paid for making application development platform agnostic and bring the same application to users independent of the terminal. Here, a solution to the speed problem could be the possibility of embedding the JVM into hardware like with Blackberry OS. On the other hand, one can argue that first java is not platform agnostic, as with several major OSs (iPhone e.g.), and second the price of living with the complexity of using JVM is simply too high, there are other solutions with better performances (cf. next two subsections).

4.2 Python Mobile

Python is an ideal prototyping tool since it is easy to learn and time efficient as it is possible to save considerable time during program development. Different Python versions exist depending on the mobile OS. The one that we will concentrate on in this paper is PyS60 running on Symbian. Usually Python scripts are much shorter than the equivalent of C, C++ and Java programs due to several reasons (Python Documentation, 2009):

- The high-level data types allow you to express complex operations in a single statement;
- Statement grouping is done by indentation instead of beginning and ending brackets;
- No variable or argument declarations are necessary.

The libraries provided by PyS60 can be divided into broad categories:

1. Built-in libraries
2. Dynamically loadable libraries

Since PyS60 is essentially an extension to the standard Python, it is known as the Python for S60 extension. Hence the categories can also be called built-in extensions and dynamically loadable extensions. The built-in extensions again are divided into two modules, e32 and appuifw (Python Mobile, 2007).

The e32 module provides access to those services that are not available in standard Python libraries: timer service, system information and listening to drivers. The appuifw module contains all the UI elements of the S60 platform. There are 16 dynamically loadable extensions. The most commonly used among them are (PyS60, 2009):

- Graphics, which provides access to the graphics manipulation capabilities of the S60 platform including loading, saving and resizing of images
- Messaging provides access to the messaging capabilities of the S60. It includes SMS and MMS services
- The inbox module, which allows one to access the inbox of the mobile on which the application is being executed. The information one can access includes the message, time of message, the address of the sender and so forth
- The camera module, which grants access to all the functionalities of the camera. This includes image modes, flash modes, maximum zoom available, and so on
- The audio extension, which provides access to the audio capabilities of the device. The functionalities include playback of different formats of audio files, recording of voice, text to speech and so on
- The calendar module, for using the calendar functionalities of the device

- The contacts module, through which one can access the contact list in a device. The contacts are provided in the form of dictionary

PyS60 is ideal for fast prototyping with no access restriction to local functionality; there are few disadvantages as the interpreter needs to be installed with an application or as separate package and slow execution speed because of the interpreter (Multi-Language Programming, 2008). Some other disadvantages of Python are some major operating systems, like iPhone, Android, Blackberry and Windows Mobile are not officially Python enabled, and in the Python environment the developer has limited access to native hardware components.

4.3 Qt

Qt is a cross-platform application framework and it runs in desktop OS as Windows, Linux and Mac. On mobile phone Qt run on Symbian and MAEMO. Qt was originally created by Norwegian Trolltech and acquired by Nokia in June 2008 as stated earlier in this report.

Qt provides an intuitive C++ class library with a rich set of application building blocks for C++ development. Qt goes beyond C++ in the areas of inter-object communication and flexibility for advanced GUI development (Qt, 2009). Since 2005, Qt had a fast development phase which makes it one of the main mobile program languages of nowadays.

In 2005, Qt 4.1 was released which consisted of basic features and introduced a tiny integrated SVG support. In the next year, in 2006, the windows vista support (which was further improved in 2007) was launched as well as native CSS support for widget styling and new 2D graphics framework. In 2008, Qt 4.4 already included concurrency framework to ease the development of multi-threaded applications, an IPC framework with a focus on shared memory, and WebKit integration. The latest release version includes QtCreator which allows fast application development and improved integration with WebKit (cf. section iPhone). The QtWebKit provides a Web browser engine that makes it easy to embed content from the World Wide Web into a Qt application. At the same time Web content can be enhanced with native controls (QtWebKit Module, 2009).

While the slogan of Java is "Write once, run everywhere", the slogan of Qt is "Write once, compile everywhere". This makes Qt to be highly platform agnostic. Compared to Java, Qt has the advantage of being compiled directly to the OSs and hence it does not need any 'translating layer' like JVM. This solves the speed and complexities, which have been connected to JVM. Another important factor is the huge investment and focus from Nokia on Qt that is a heavy argument for seeing it to be a valid competitor with other software development tools.

5. Conclusion

There are a number of different business models on the mobile applications and services market. In the recent years, however, we are witnessing a shift towards more device centric models, where device manufacturers go beyond the terminal market and take share in the value creation at services and content level. The aim of this paper has been to focus on this model and try to identify the strengths and weaknesses of different platforms through a techno-economic comparative analysis of the major newly emerged mobile business models.

The Apple/iPhone model may be seen as the initiator and trendsetter for the device centric models that all major platforms in a way or other have entered to. The model is that an ‘application store’ is launched and the capability for accessing this store is implemented into the devices. For majority of the available models also development kits are made available so third party developers can build new application and upload them to the store. In short, the mechanism of many of these models is that they launch a new market place, where producers and consumers meet and trade for applications and services. The revenue split between the platform provider and developer of the services follows different models and rates. This has resulted in a number of different initiatives like Apple’s ‘App Store’, Google’s ‘Android Market’, Blackberry’s ‘App World’, Nokia’s ‘OVI Store’, Palm’s ‘Palm App Catalog’ and Windows Mobile’s ‘Windows Market place’.

With regards to mobile operating systems, Symbian has for long time been the dominating technology, however, it seems that in the transition to the smartphones other operating systems like iPhone and Android are taking the lead. In the smartphone market some of the operating systems like iPhone and Blackberry are tightly connected to the business logic of the platforms, with reasonable consumer bases. Hence it is difficult, at least in the near future, to see a winning operating system on the market. However, the Google/Android initiative of developing an OS which can run on all mobile devices is important and interesting to follow in the future.

With regards to the software platforms, Java (J2ME) has been the far dominating platform for mobile devices. However, it has been heavy and slow to work with. In the recent years Qt is getting more attention and focus as it is highly platform – agnostic and does not have the extra layer of complexity of, e.g., JVM in Java.

In the paper we analyzed different operating systems with their corresponding business models and the deployed software development platform. The results are depicted in table 2.

Parameter	iPhone	Android	Symbian	Blackberry	Windows Mobile	WebOS
<i>Openness:</i>	*(*)	*****	****	**	***	*****
<i>Look and feel:</i>	****(*)	***(*)	***	***(*)	**	*****
<i>Web integration:</i>	*****	*****	***	**	***	*****
<i>Industry and community support:</i>	****	****	****	**	***(*)	*(*)
<i>Future perspectives:</i>	***	*****	**** ⁸	**	***(*)	**
<i>Deployed software development</i>	Objective C, WRT widgets	Android Java, Native	Qt, Python Mobile, Flash Lite, Ruby,	J2ME, WRT widgets	J2ME, Qt, Visual Basic, .NET, Visual C++, WRT	WRT widgets

⁸ Assuming Nokia and Samsung will not discontinue their work on Symbian, despite the rumours of lack of support in the coming years.

<i>platform:</i>		C/C++/ ARM ⁹	.NET, WRT Widgets, Symbian C++, Standard C/C++		widgets, Flash Lite	
------------------	--	----------------------------	---	--	------------------------	--

Table 2: Comparison of major Mobile Operating Systems

6. References

ANDERSON P. (2007): "What is Web 2.0? Ideas, technologies and implications for education", JISC Technology & Standards Watch

BURKHARD, S. L. (2009): "Economies of Application Development Programs", in "Internet Economics IV", Edited by Burkhard Stiller, Cristian Morariu, Peter Racz & Martin Waldburger, University of Zurich, department of Informatics, Feb 2009

FEIJOO, C., PASCU, C., MISURACA, G. & LUSOLI, W. (2009): "The Next Paradigm Shift in the Mobile Ecosystem: Mobile social computing and the increasing Relevance of users", COMMUNICATIONS & STRATEGIES, no 75 (3rd quarter)

FEIJOO, C., MAGHIROS, C. & RAMOS, S. (2008): "A survey of firms in the mobile content and application ecosystem", paper presented in the 19th European Regional ITS Conference, Rome, Italy

HEATLEY, D. & HOWELL, B. (2009): "The Brand is the Bundle: Strategies for the Mobile ecosystems", COMMUNICATIONS & STRATEGIES, no 75 (3rd quarter)

HOEGG, R., MARTIGNONI, R., MECKEL, M. & STANOEVSASLABBEVA, K. (2008): "Overview of business models for Web 2.0 Communities", St Gallen university, Institute of Media and Communication Management

SCHEIBLE, J. & TUULOS, V. (2007): "Mobile Python: Rapid prototyping of applications on the Mobile Platform", SYMBIAN, Wiley, 2007

IDATE (2009): "Digital World Yearbook 2009", IDATE annual publication

KHANA, A. (2009): "The future of Mobile Application Storefronts", Wireless Expertise market report

KILGO P. M., (2008): "Android OS: A robust, free, open-source operating system for mobile devices", Paul Michael Kilgo, University of Alabama, available on: <http://pk-fire.com/etc/wtf/android-os-final.pdf>

O'REILLY, T (2005): "What is web 2.0: design Patterns and Business Models for the next generation of software. Available on line at: <http://oreilly.com/web2/archive/what-is-web-20.html>

POUWELSE, J.A., GARBACKI, P., EPEMA, D. & SIPS, H. (2008): "Pirates and Samaritans: A decade of measurements on peer production and their implications for net neutrality and copyright", Telecommunications Policy, volume 32

⁹ Assembler for the ARM processor architecture

6.1 WEB links:

(9to5mac, 2009)

<http://www.9to5mac.com/android-rom-install-256-droid>

(App Catalog, 2009)

<http://pdnblog.palm.com/2009/08/developers-bring-us-your-apps-and-sell-them-too/>

(Ars Technica, 2009)

<http://arstechnica.com/microsoft/news/2009/08/windows-mobile-loses-27-of-smartphone-market-in-q2.ars>

(Ballmer, 2009)

<http://news.idg.no/cw/art.cfm?id=F2F7C35E-1A64-67EA-E4BC04F120F0B898>

(Bloor, 2009)

http://www.allaboutsymbian.com/features/item/WRT_Another_arrow_or_killer_weapon.php

(DigiTimes, 2009)

<http://www.digitimes.com/news/a20091111PD205.html>

(Gartner, 2009)

http://www.computerworld.com/s/article/9139026/Android_to_grab_No._2_spot_by_2012_says_Gartner

(Gizmodo Windows Mobile, 2009)

<http://gizmodo.com/5374876/windows-mobile-65-review-theres-no-excuse-for-this>

(Gizmodo HTC, 2009)

<http://gizmodo.com/5400593/htc-touch-hd2-review-a-tragedy>

(Hamblen, 2009)

<http://news.idg.no/cw/art.cfm?id=F2F7C35E-1A64-67EA-E4BC04F120F0B898>

(Helmreich & Doriot, 2009)

<http://blog.seattlepi.com/microsoft/library/20090930smartphone.pdf>

(Nokia, 2009)

<http://thereallymobileproject.com/2009/11/nokia-dropping-symbian-from-n-series-by-2012/>

(Nokia deny, 2009)

<http://thenokiablog.com/2009/11/18/nokias-response-dropping-symbian-nseries-2012/>

(Samsung deny, 2009)

<http://news.softpedia.com/news/Samsung-Will-Still-Support-Symbian-126925.shtml>

(Symbian, 2009)

http://developer.symbian.org/wiki/index.php/Roadmap_for_the_Symbian_Platform

(W3, 2009)

<http://www.w3.org/DOM/>

(Windows Mobile 7 preview, 2009)

<http://www.windowsmobile7.com/hot-windows-mobile-7-insight/>

(Windows Mobile 7 trailer, 2009)

<http://www.windowsmobile7.com/windows-mobile-7-ui-preview/>

(Zune, 2009)

http://www.youtube.com/watch?v=r6m-n09LjB8&feature=player_embedded

<http://gizmodo.com/5360126/zune-hd-review-the-pmp-evolved>

(Zune Windows Mobile, 2009)

<http://www.electricpig.co.uk/2009/11/16/microsoft-puts-zune-mobile-plans/>

(CLDC, 2009)

Sun Microsystems, Connected Limited Device Configuration (CLDC)

<http://java.sun.com/products/cldc>

(J2ME, 2007)

The Java ME GUI APIs at a Glance

<http://developers.sun.com/mobility/midp/articles/guiapis/>

(Mobile Programming, 2007)

Fitzek and Reichert

Mobile Phone Programming and its Application to Wireless Networking

<http://www.springerlink.com/content/978-1-4020-5968-1>

(Multi-Language Programming, 2008)

Bernd Wiegmann

Multi-Language Programming

http://developer.symbian.com/main/downloads/papers/Multi_Language_Programming_Part1_Overview.pdf

(PyS60, 2009)

Mobile Programming in Python using PyS60

<http://www.devshed.com/c/a/Python/Mobile-Programming-in-Python-using-PyS60-Getting-Started/1/>

(Python Documentation, 2009)

Python Documentation

Release 2.5.4

<http://www.python.org/doc/2.5.4/>

(QtWebKit Module, 2009)

<http://chaos.troll.no/~tavestbo/webkit/domapi/qtwebkit.html>

(Qt, 2009)

<http://qt.nokia.com/files/pdf/qt-4.4-whitepaper>