The 8th International Scientific Conference
eLearning and software for Education
Bucharest, April 26-27, 2012
10.5682/2066-026X-12-172

ARE MOBILE LEARNING DEVICES READY FOR ENTERPRISE DEPLOYMENT?

Mugurel PATRICHI
"Alexandru Ioan Cuza"Carol I blvd, Iasi Romania
E-mail: mugurel@feaa.uaic.ro

***Abstract**: One of the strong points of mobile learning is considered to be the ability to deploy in corporate environment, for a "learn anywhere" approach. But it is easy to see how employees, once provided with these devices, will want more from them. We consider, therefore, that an organisation analyzing the possibility of deploying personal devices for training and on-the-spot assistance, should consider the fact that the employees will want more than to run the training software. It might be something simple as accessing email or going online. But in a corporate enviroment, these activities should be regulated, not only as a good practice, but for compliace with standards or laws. This paper looks beyond the multimedia capabilities of these sort of devices, and wonders if the operating systems of the smartphones and tablets are fit to access the organisation's network and secure areas*

***Keywords:** mobile learning, organisation, mobile training, security policies*

I. INTRODUCTION

One of the touted abilities of mobile-learning is the ability to learn anywhere anytime. Among the pillars of mobile education we can find "personalization of content", "education available anywhere", and context-aware information [34]. As author and consultant Clark Quinn puts it *Mobile Learning is not* (only) *about courses on a phone. […] while there are learning implications for mobile devices, it's really about performance support […] the real opportunities are providing context-sensitive support for the mobile workforce.*

Of course, not everyone agrees this qualifies as "mobile learning", but as mobiles terminals are used to access information right when it's needed, managers might come to ask whether isn't appropriate to invest in putting this type of context aware information at the fingertip of their employees. Whether we called it e-learning or not [referinta la articolul meu], an employee who has access to information at the workplace (be that on the office, in a business trip or on top of an oil rig), when he needs it, can perform better. In the not so distant past, at this point it should have been appropriate to discuss the hardware investments that an organisation should do in order to make sure every employee has the terminal required for this endeavor. But as the yearly sales of terminals soar way beyond the 50 millions, one should ask if it's not appropriate to use employee's own terminals as the receiver, with the enterprise providing the content and the support. Furthermore, studies have shown that even in organisations where the management is not considering providing support for mobile terminals, the employees are actively lobying for it ([111],[143])

We should mention from the beginning that there are two ways an organisation could implement this type of mobile learning software: using native applications, or using web applications. The current trend is for web aplications [alege 166, 57,88], using Rich Internet Aplications like Flash, Java, .Net or HTML5, but one should not ignore the native software, which still has advantages like better hardware compatibility, better memory management or, in some cases, more secure access to the terminal (alege [101],[74],[22])). While the common denominator in providing internet-based

applications is that it is user's job to secure his connection to the internet, we side with those who consider that it is bad practice to leave securing organisation's assets to the users.

For this paper we will consider the case of an organisation who would like to allow employee-owned terminals to their organisation networks. The findings should also apply to organisations that provided smartphones or tablet for their employees in the recent past, but would like to extend their usage without buying new ones (especially if the equipment isn't depreciated yet). In theory, this should be the equivalent of using the laptop to connect to the corporate network. In practice, we investigate if mobile terminals (smartphones and tablets) do behave in the same manner and be treated as laptop equivalents.

No matter who is actually assembling the terminal, it is the operating system the one that decides what applications can be run on that terminal. However, an analisys of the operating systems from the point of usability for the learning process could very likely lead to an enumeration of the applications available for the respective terminal. But such an analisys ignores the compatibility of the terminal to the business enviroment, more exactly its compliance with the security policies and its capability to accept the management of a senior entity (the department tasked with information security in the organisation, usually the IT Department). In this paper we've tested the three top operating systems for mobile equipments (smartphones and tablets) in order to evaluate their capability to comply with the requirements of a business enviroment: the acceptance of security policies, remote management policies, and other features related more to data protection and exchange then to Facebook plug-ins or multimedia capabilities.

We annalised the following operating systems (between brackets we mentioned the terminals themselves we tested):

-Android (LG Optimus, Sony Ericsson Xperia and Samsung Galaxy Tab 8.9)
-iOS (iPhone 3G S, iPad2)
-BlackberryOS (BlackBerry Torch 9800, BlackBerry 9360 Curve)

The test itself consisted on acessing the management features. Unfortunately, we did not receive permissions for installing applications or applying techniques similar to Jailbreaking. Also, we could not test the Blackberry Server. We consider we overcomed these shortcommings by extensively researching how these features are covered on the specialised websites and filtering the content relating to our endeavour. For most part we ignored the multimedia and aestetic features of the terminals. We analysed not only the latest models, but also older versions of the same operating system, for two reasons:

-on one hand, in many cases, the latest versions of the operating systems do not run on older terminals (i.e: Blackberry, Windows Mobile). In other cases, some versions run on the tablets, not on the smartphone (Android)
-on the other hand, there are still terminals on sale that are using older versions of the operating systems. Android is the prime example here, although we can find Blackberries with versions 6 and 7 on sale.

The internet and main-stream mass-media are filled with reviews for these equipments, but since they where marketed more like entertaining devices, these reviews do not focus on the security and policy compliance. Especially when documenting the older versions of the operating systems, the media covered extensively the multimedia abilities, the external design features and the hardware capabilities, with little-to-no coverage of the security policies. Because of the speed this technology evolves, it is hard to find books addressing the issues of configurating these terminals for enterprise development. However, it should be noted that the user manuals do cover setting up security policies, and there are books about the APIs the programers can use to develop aplications.

II. BLACKBERRY OS

When the smartphone edition of the Blackberry was launched in 2003 by the Canadian firm Research in Motion, it was a prime example of convergent communication technologies. The operating system, Blackberry OS, allowed push-email, internet faxing, internet browsing, alongside to

posibility to phone and send text message. Depending on the type of the terminal in discussion, the finger-friendly qwerty-keyboard was augmented by a scroll wheel, a mechanical trackball or a digital one. The latest versions natively support Java and there was an actively encouraged partnership with Adobe to suport Flash natively, allowing easily development of aplications for third-parties. Probably the strongest point was the wireless syncronisation with Microsoft Exchange and Lotus Domino, for collaboration tools, calendar and email. The last Blackberry OS version for phones, 5.0, which was introduced in September 2010 expanded the native suport for Java, allowed full activation and syncronisation with Microsoft Exchange Server (e-mails, scheduling, tasks, notes, contacts) and also with Novell GroupWise and Lotus Notes (groupware and instant messaging for organistions).

Blackberry achieves this conectivity by using a middleware aplication called Blackberry Enterprise Server, which is, basically, doing all this email redirecting and information syncronisation between servers, desktop computers and mobile terminals. Beside the server itself, RIM launched a free application package called Enterprise Server Resource Kit, that can be used for:

-remote management for terminals using command line, allowing for scripts that can automatise some actions that would otherwise require a lot of mouse clicks

-logs analisys

-monitoring and recording of user activity

-monitoring and recording of the data stream between users and corporate servers, including statiscs and metrics

-monitoring and auditing of the push-email system

There is an alternative to the implementation of a server, the Blackberry Internet Service. This service is provided by the telephone carriers in 91 countries, but it is actually managed by RIM an organisation that do not want to implement a BlackBerry Enterprise Server can provide access to Blackberry services by contracting a dedicated plan from the carriers. Another free option is the Blackberry Enterprise Server Express, which does not rely on carriers, but only requires internet access. In both cases, there is a problem with data jurisdiction; although the user access data (i.e. emails) from a local server, data is actually routed to RIM servers in Canada and then rerouted through carrier's system/broadband to the terminal. Unfortunately, this means that when something goes wronmg at the RIMS's servers, all of the syncronisation stops, and unfortunatelly, this happened before ([41]).

All the above-mentioned security options where featured in the fifth incarnation of Blackberry OS and they surpass by far the other platforms in term of busineses-oriented tools. The sixth version improved on multimedia and social features (Facebook, Twiter, social tools for use with other Blackberry users).

In august 2011 RIM introduced Blackberry OS 7, which allowed for separation between company data and private data (including the posibility of separate remote control policies). However, this version was incompatible with the previous Blackberries, in other words, existing Blackberries could not be upgraded to make use of the new features. Even so, there were (and this still aplies) few applications developed for Blackberry 7, as some developers resented the countless aprovals that they had to receive in order to get their software to the potential clients ([40]) .

With all these business-oriented tools and technical features, it might come as a surprise the fact that Blackberry is failing fast behind competition. However, we agree with those who separate the potential clients in analists, and consumers. ([42]). And while analists might be impressed with what Blackberries can do, consumers are not: Blackberry Storm, the first touchscreen smartphone launced by RIM (2008) has been criticised ([165]) for its bulky appearance compared to its iPhone rival, lack of wireless capabilities and a barely functional onscreen keyboard. However, it still sold 1 milion pieces in the first week. Storm 2 was launched a year later and was described as "doing basically the same things, only better"([19]). The next touchscreen terminal, Torch, launched in 2010 as an "iPhone killer" came with a sliding keyboard, but without any sort of processing power upgrade; it had a 5 MegaPixels camera, but no 4G support and, more important, a low-resolution 480x360 screen ([20]). RIM decided to keep producing no-touch-screen terminals (the Bold serie), which had the same features that Blackberry OS 6 and 7 have to offer for their touch screen counterparts.

In 2011 RIM attempted to launch a tablet under the Blackberry brand, called Blackberry Playbook. The tablet came with some advanced multimedia features (multitasking, powerfull onboard

cameras, wirelless file transfer) and also advanced suport for . Furthermore, policies and customisation set for corporate Blackberries were automatically implemented on the Playbooks. However, Playbooks did no have native support for email, adress-book and calendar syncronisation. In order to achieve this, one had to install a software that would connect the Playbook to a Blackberry, through which all this syncronising was be made. Furthermore, when compared to the competition, Playbooks did not come with out-of-the-box software for videochat or GPS, and no mobile internet. The tablet tanked in sales, but in February 2012 a new version of the Blackbery Tablet OS was released, fixing all these issues.

Organisations willing to invest in Blackberry smartphone support should take into account the relative small number of aplications developed for their operating systems. Still, there are antiviruses for BlackberryOS and, as mentioned before, out-of-the-box aplications for management. The Blackbery Tablet OS 2.0 bought the posibility of running applications developed for Android. As of the time of writing there are only around 10.000 aplications in Blackberry Market, but an Blackberry user could also install aplications from Google Play (former Android Market) at his own risk.

III. iOS

iOS is the operating system for iPhones and Ipads, made by Apple Inc.

If we take into account the upfront-stated focus on entartainment, it should not come as a surprise that there are few business-oriented tools (ot-of-the-box or third party) for iPhone and iPad. One reason for this can be found in Apple's constant refusal to allow developers access to the kernel of the iOS, in order to develop core-level applications. We should note, however, Apple's improvements on these issues compared to the first iPhones. Unfortunatelly, the process is a very slow, the multimedia and entartainment features taking priority.

When compared with other operating systems developed for smartphones (i.e: Blackberry OS), iOS shows greater space for improvements (functionality-wise): firstly, it lacks a general management application (like the management consoles in Windows or their equivalent in BBOS). While the fourth version of iOS (introduced in April 2010) has a Configuration Utility, it only allows access to Certificate-based authentification and encription mecanisms, which we consider to be very little. The fifth version of iOS (introduced in september 2011) brought little out-of-the-box feature for enterprises: IT Departaments can now prevent sending of corporate mail to private accounts, can enforce using S/MIME mail encryption and can prevent usage of untrusted certificates).

The lack of out-of-the-box tools for terminal managements can also be illustrated by looking at the finnancial success of third-parties willing to develop aplication for iOS-powered terminals (i.e. MobileIron). However, Apple's closed policy regarding its operating system (including update schedule) and the lack of developer access to core code leads to situations when security updates mess all the customisation that the organisation made to the terminal through the use of these third-party application.

iOS 3- powered terminals had another issue with the security patching process: the users had to receive these patches through email (or by clicking a link). This put the entire responsibility for security updates in the hands of the user, with no enforcement tools for the IT Department. Again, this was corected in iOS 4 (now the equipment can be set to accept automatic updates from Apple, similar to Windows Authmatic Updates for the PC), but is still a long way from Blackberry's Enterprise Server that can provide business enviroment tested updates. Especially since iOS5 cannot be installed on the first generation iPhones.

While this updating issue can be easely dismissed as some IT Department "fuss", we consider it to be really the tip of an ugly iceberg: ensuring compliance with diferrent security and auditing standards (and we include here those standards related to the information flux inside an organisation). While one could find on the internet guides for ensuring iOS-powered terminals a minimal compatibility with appliable standards, (i.e. [168]) we consider the lack of centralised tools to implement, change or enforce security settings as a serious security problem, if we take into account the great responsibility placed on the IT Department by laws like Sarbans-Oaxley etc. Furthermore,

some basic questions that an auditor could ask are: is the software capable of providing data regarding all the user activities? Can the IT Department provide a list of phone calls that the user made on the organisation-provided terminal? Or a list of messages sent, or the sites accessed through the corporate network? Or the location of the phone when it was last time unlocked? The iOS software can provide very little of this information, and is certainly no match for Windows or Blackberry logs.

All of these issues force an organisation willing to support an iOS implementation to contract a third party developer to correct these shortcomings. Besides the extracosts, there is still the issue of Apple's lack of comunication with the developers. We already stated that third-parties have access to a very little list of tols to create programs with and even shorter means to reach the kernel. Furthermore, Apple restricts the installing of applications to its Apple Store (which, in turn, creates another way of controlling the apps available for iOS). The problem here is that ussualy the IT Departments have support-related activities, and more-often-than-not do not develop in-house aplications. Thus, IT depends on external developers to respect a set of business-specific objectives, but also a set of external requirements as specified in laws or standards. Thus, in the end, these outside developers are the only ones that are managing the relation between the organisations and the terminals, but it is the IT Department alone that it is held responsible in case an audit fails. And we consider that an audit has a lot of chances to fail since, as someone put it, with all the propaganda of being ready for enterprise deployment, what users do on an Apple terminal is an opaque discussion between the user and Apple. It's a shame, but Apple and their partners know more about your users then you.

The…cold relationship between Apple and programers it is limited to iOS. There are a lot of open-source applications bundled with the operating system. Thus, any breach in an open-source module cand become public and exploitable befor Apple can release a security Patch. On such example is the on-board browser, Safari, which is based on an open-source engine called WebKit. This particular software is a favorite at hacking contests like Pwn2Own ([68],[46]). Another issue that an organisation willing to support an iOS-wide implementation is the restriction that Apple puts on the developing tools accepted for software development:

-Java does not work on iPhone/iPads, ([141]) and the late Steve Jobs said it will never do. ([153]) Although Java's owner, Sun, announced in 2008 the intention to develop a way through which Java-compiled code to pe run on iOS, the main issues where legal ones ([71]). After Steve Jobs (which was known for his stubberness) passed away last year, users hope this restriction will be lifted

-.Net does not work out-of-the-box, although there is a thrid-party application called MonoTouch that can take .Net code, compile it into its own code and then can run this code as itself, thus bypasing Apple legal restrictions([95]).

-Flash is not supported, neither as executable, nor as multimedia webcontent on sites such as YouTube. Apple claimed that Flash content is to dificult to use in iPhone enviroment ([123]), but did't comment on why is not available on iPad. Starting with 2009 flash's developer, Adobe, offered a way to export the applications created through Adobe Flash Professional CS5 in an iPhone format ([10]), which would allow the flash developers to export their aplications into an Apple Store approuved format. Still, any attepmt to run flash web applications or flash multimedia content inside iOS' browser, Safari, fails.

Lacking access to the core level of the OS, the programmers cannot develop anti-malware or intrusion-prevention application for iPhone and iPad. This situation alone brings the two devices outside any security policy implemented at an organisational level. Existing third party aplications can, at most, help a central operator implement some Apple-established policies in all the terminals. For instance, using MobileIron one can check on all terminals the option of erasing the memory after ten consecutives password fails. But as of now there is no antivirus software for iOS, and, more troubling, the biggest antivirus developer for businesses, Symantec, prides itself with the fact that its aplication for iOS does just that: management. Nothing about preventing or curing viral infections ([16]). The same goes for another security application for iOS, Sophos ([161]).

Besides these software problems there is a hardware issue that an organisation willing to support or implement an iOS has to take into account: there are no service providers. Everything has to be done by Apple. Even replacing a battery requires sending the whole terminal to Apple. Because in most organisation every IT-related problems is to be solved by the IT Department, any hardware problem puts a pressure on the human resources in this department, but also a pressure on the firm

itself to come up with a solution if the person owning the phone depends on it to solve some critical task.

All these issues add up to a rather negative image of iOS. And although iPhones and iPads spearheaded the delivery of multimedia content on portable devices and look like the best tool for any sorof mobile learning implementation, from a compliance point of view we cannot consider these terminals as suited for deployment or support inside an organisation security enviroment. An organisation willing to accept them will have to invest heavely in third-party development of applications and in securing a network enviroment that do not depend on the terminal for even the most basic security requirement, an antivirus.

We cannot end our annalisys without mentioning the jailbreak, which is a process through which an iOS-powered terminal can be unlocked, thus lifting any restriction Apple set for installing applications only through its Apple Store. While legal in United States ([79]) and in the European Union ([139]), this process uses freelly available applications downloadable from the interet that use exploits in order to access the root level of the iOS (thus jailbreaking is a form of hacking [56]). Once the terminal has been unlocked, the user can download software and customisation files that did not go through Apple's aprouval process, from dedicated aplication stores (like Cydia). These programs are full commercial applications (including client suport), not hacked Apple software. They cover a wide range of posibilities from interface customisation to playing multimedia files that iOS claims it cannot play ([55]).

A jailbreaked terminal keeps its basic functions (make phone calls, download and install applications from Apple Store) but loses any form of waranty and support from Apple. It is also a fully reversible process (through iTunes) [112], but the security breaches it uses are constantly patched by Apple. Unce a security patch is in place, the developers of jailbreaking software find new security holes to allow users access to root level. We consider the fact that they have been doing this for some times another sign that the security that Apple claims it ensures to its terminals needs revising

IV. ANDROID

Contrary to Apple's tigh control on the application that iOS can run, Google announced that it will not develop many applications for its operating system, Android. Instead, Google developed the Android as a platform and released it for free (open-source) to the developers, to customise it, adapt them for their terminals and, of course, create programs. This is simoultaneous oportunity and risk: opportunity, because the developers have greater access to the kernel and they can program applications better suited to an organisation needs. And one of the best way to test this is by looking at the suite of applications developed for (remote) management of terminals: the administrator have more security tools at their disposal, including, as the spearhead of any security policy, antivirus software (a list can be found here [90]).

There is also a risk in this open approach. Google does not check all the application in its Android Market (now rebranded Google Play) and it wouldn't be the first time users download malware application desguised as legit software. Two times, in 2010 ([25]) and 2011 ([24]) Google announced it deleted some applications from Android Market ad it made use of a special features in Android terminals that allows an application to be remotedly unninstalled at Google's command.

In Android case we can talk about a fragmentation phenomenon: the license allows terminal producers to alter the operating system without notifying Google or the community [102] Thus, software that was design for Android might not run on certain equipments (especially in USA, where carriers have exclusivity on selling terminals). This is the reason the operating system cannot be instantaneous updated (like iOS), but one has to wait for the carrier to patch Google's update and make it compatible with the OS alterations and with that device hardware particularities. The sole existence of these carrier-made customisations can lead to compatibility issues especialy when an organisation would like to develop applications that will make use of employee-owned terminals, as it can pose problems in terms of compatibility.

We cannot call Android versions before 2.2 fit to be used in business enviroment ([83]). Android 2.2 "Froyo" (launched in May 2010) brought support for security policies and complex passwords, remote terminal lock and remote memory wipe for stolen or missing equipment. Android 2.2 also introduced complete back-up and restore, of data, even remotely [126]. Application-wise, Froyo allowed software installs to be made on external memory and improved memory management. For organisations willing to implement web applications, Froyo speeds up JavaScript loading up to five times previous versions [127]). With Android 2.2 there is the posibility of updating all the installed applications through a single click, and error messages can be forwarded directly to the application developers, without the need to go through the IT Department.

As of March 2012 Froyo is placed second in Google's Ranking of Google Play access ([162]). Froyo continues to be used in budget tables, although it wasn't designed to take advantege of a bigger screen.

The next versions, Android 2.3 -Gingerbread and Android 3.0-Honeycomb (last one exclusively on tablets) focused more on technical and multimedia features and less on business-oriented features. Still we have to mention the improved battery life (for organisations that will implement internet-based software which is heavy on battery usage) and in Honeycomb's case, improved suport for file encryption and password enforcing. Android 3 security features were brought to smartphone by the fourth version, IceSandwich, but vecause the updates need to be confirmed first with the terminal manufacturers, smartphones sill have to wait to get the update.

We cannot end this analisys without stating that, because of the relaxed policies, one can take advantage of Google Play and install malware. If the company chooses to use employees' terminals and just invest in software, there is little in terms of what can impose on a employee regarding what it is allowed or not to do with his own terminal. However, we should note that the Android operating system is built to ease development of application.

V. CONCLUSIONS

We consider that the present day mobile terminals can be used to deliver personalised date and context aware information, making them prime candidates for education anytime anywhere. However, it becomes clear that securing these terminals is not something to be left at users' discretion.

At this moment the one operating system that trully offer state of the art security for corporate data accessed through mobile equipment, Blackberry, powers some devices that are considered by many to be unappealling and obsolete in both design and features, coivering only xxx percent of the market. However, we side with those who consider the Playbook tablet very appealling in both features and security, but this appeal does little for an organisation that want to use existing equipments as the measn for delivering content.

The firm that started all, Apple, keeps a way too tight lock on its kernel code to allow for even the basic of protection, an antivirus software, and an even tighter lock on user data to allow some basic auditing. And although we found guides detailling what applications to install and what configurations one has to apply in order to make the terminals compatible with standards and laws, management has to ask if it's really worth it. Of course, the organisation could place the whole responsibility in the hands of the users, thus blaming them for and if anything happens. They would be covered legally. Everithing lies in the data that gets lost, or stolen, or missplaced.

The market leader, Android, offers the possibility of best of both worlds: it allows developers to develop applications that could secure a terminal as best as possible, while delivering the multimedia experience that users loved in Apple's products. While this is certainly more flexible in terms of what it can be managed, it should be noted that most offerings do not come out-of-the-box, and required a third-party separate bill.

We do believe that in the near future producers will try to capitalise on user desire to use their terminal in an enterprise enviroment and will address these issues.

References

Reference Text and Citations

References should be set to 9-point, justified.

[1] John, I., Henry, A., 2009. Paper elearning. *In ELEARNING 2009, 6st International Conference on Elearning*. MITUS Press. Pag 23
[2] Doe, J., 2005. The book, The publishing company. Amsterdam, 4th edition. Page 123