

The Open Source Cellphone :
Lead Users and Early Adopters in the Mobile Phone Industry

Anmol Madan
Media Lab, MIT
anmol@mit.edu

Class Paper 1
15.351 – Innovation Management and Strategy

Abstract

In this paper I propose that disruptive applications and business models for mobile phones in North America are limited not by hardware or software capabilities, but by a restricted application development environment and the inability of heavily invested incumbents (service providers / carriers) to capitalize on such radical innovation. I provide examples to illustrate how a new entrant with an open source mobile platform could effectively harness lead users as developers to create disruptive applications and capture significant market share.

US Mobile Phone Industry in 2006

The mobile phone will soon be the most popular consumer device in the planet. Increasing mobile phone penetration is a global phenomenon with approximately 200mm mobile phone users in the US, 400mm in China and 80mm in India last year (up 47% in just one year). Household penetration of mobiles is highest in East Asia (Japan, Korea, etc.), followed by Europe and then North America. The number of mobile phone subscribers in India, South America and China exceeds fixed-line users and more people in Mexico checked their email on mobile phones than on computers last year.

It is also evident that cell phones have driven a remarkable cultural transition. About half of the 800 mm handsets sold last year were high end smart phones, with features like in-built cameras, multimedia playback, email, web access and external memory. Even in developing economies like India and China, listening to music on phones, snapping videos at social events or sharing those videos with friends over bluetooth is commonplace. South Korea is the global leader in mobile entertainment, setting standards for satellite video broadcast to mobile phones.

The North American mobile phone industry is unique in the extent of control exercised by the service providers (or carriers) and to some extent the phone manufacturers, over end consumers. With an average handset lifetime of 3 months, handset manufacturers like Nokia and Motorola survive by innovating rapidly on styling and functionality. The service

providers are incumbents who have invested heavily in building a mobile phone network and sometimes stand to lose from radical innovation. For example, in North America and parts of Europe, one of the crucial factors is VOIP (Skype or SIP) on mobile devices, especially for long-distance voice calls. While some service providers in North America and Europe have accepted and even encouraged VOIP calls on their 3G data networks, almost all carriers are averse to consumers using public 802.11 wifi on handsets to make phone calls. Most phone manufacturers in the US competitively negotiate handset deals with carriers. Consequently, Nokia and Motorola have been struggling to release mainstream wifi 802.11 phones in the US. A broad comparison between current carriers and a potential new entrant is done below.

	Strengths	Weaknesses
Incumbents: Traditional carriers	1) Own high-reliability GSM bandwidth 2) Have financial leverage because they interface with end customer and control billing 3) Leverage with handset manufacturers (direct sales to consumers)	1) Heavily invested in having consumers use that bandwidth 2) Can not “open” their platforms completely because users would switch to the cheapest communication network (not always theirs)
New Entrants or MVNOs	1) Not tied to the traditional business models 2) Take advantage of cutting edge technologies – VOIP over wifi / bluetooth, wifi 802.11 mesh networks etc.	1) Can never replace the carriers completely 2) Will almost always rely on GSM connectivity for the last 10% 3) Limited resources

Why Open Source Cellphones?

The mobile handset industry is like the PC industry in the late 80's and early 90's – many competing standards and manufacturers and low interoperability. While there some standardization for application developers in Java™ and Symbian™ programming, most handsets have slightly varying implementations and developer support is encouraged only for games and simple applications. Network service providers still control how applications are distributed to customers and can rarely distribute a single application that works across all their handset models. Varying standards and programming APIs have created choke points for application developers and consumers.

As a powerful mobile computing platform with pervasive mass-market presence, cell phones have a potential far beyond voice, personalization (e.g. ringtones) and entertainment. Research and commercial groups have successfully demonstrated social networking, healthcare, education and mixed network communication (wifi, mesh and carrier based) mobile phone applications. Since the effect of such ‘vertical’ applications on existing profit streams and current competencies is unclear, very few industry incumbents are willing to commercialize disruptive applications which might transfer control to the consumers and application developers. At the same time, choke points in interoperability do not allow application developers or new entrant to readily target a large installed consumer base.

An interesting approach in such a case is the open source model. Open source lead users have a long history of innovation and *hacking* on mobile devices. Two examples are provided below :

Example 1: OpenZaurus and ZaurususerGroup

The Sharp Zaurus family is a series of QT/Linux PDAs, first released in the US in 2000. The entry-level Zaurus features a 200Mhz processor, touchscreen interface and wifi 802.11

support (with an add-on card). The Zaurus was targeted at the mainstream PDA market of¹ that time, the power executives and technology geeks. While the Zaurus was remarkably successful in Japan, it enjoyed mediocre success for a couple of years in North America, before dying down in mass market.

However, with almost no one realizing, the Zaurus soon became a huge hit in the underground Linux *hacker* community. Many technology savvy people who worked in IT departments or were familiar with Linux started carrying the Zaurus, often using it as an ultra-portable laptop replacement. Sharp initially supported the developer community in small ways (the most important -- releasing an open cross-compiler, so developers could attempt to port desktop Linux applications on their own). Sharp eventually turned a blind eye to the community after it decided to stop retailing the Zaurus in North America.

The Zaurus is no longer available in stores, but user innovation continues unabated. Websites like OpenZaurus.org and ZaurusUserGroup.org have become focal points of the Zaurus user movement. Today, two years after the the devices went off shelves, Zaurus User Group forums have 9000 active members (mostly lead users and hackers) with about 120,000 messages posted on last count. OpenZaurus, a group responsible for creating firmware upgrades to replace the original Sharp firmware, enhanced with better tools and drivers, posted its latest update fifteen days before the writing of this paper in March 2006 (compare this to Microsoft or Sharp). With the PDAs no longer available in stores, they command a high price in online stores and Zaurus User Group has detailed instructions on how to buy them in Japan and to convert them to English.

Zaurus users have shown almost a cult-like fascination for their devices. The Zaurus Software Index has become a public warehouse of over 2000 applications, created by open source developers. The Zaurus User Group has Howto's and forum discussions to greatly extend the capabilities of the Zaurus, far beyond Sharp's initial plans -- using the Zaurus for GPS navigation, to connect to external bluetooth peripherals, to periodically auto-scan for

¹ The term *hacker* is used interchangeably to describe an open-source programmer in this paper

public wireless networks, and to convert the Zaurus into a 20GB music player or streaming music server. Illustrative applications and capabilities are listed below in Table 2.

Zaurus Software Index	Zaurus User Group
Sample user-supported Applications	Sample HOWTOs & Forum Discussions
VOIP software (Kphone & Linphone)	GPS Hardware for the Zaurus
MPEG4 video playback (Dr.Z player, VLC)	Connecting a 20GB external Hard Drive and other USB peripherals (Zaurus as a USB host)
Doom (popular PC game, one of 500 other games)	
Apache-PHP web server (use the PDA to host dynamic websites)	Connecting to a bluetooth wireless keyboard
Windows Samba server (use the PDA as a Windows network file server)	Dual Booting the Zaurus PDA (multiple firmware or operating systems)
Medical Imaging viewer (televew)	Using the Zaurus to make GSM phone calls
Nintendo GameBoy emulator (GameBoyZ)	Using the Zaurus to make VOIP phone calls
Muslim Prayer Generator Tool	MPEG Encoding (converting videos)
Talking English-Spanish Dictionary for Tourists	Configuring up streaming music server
GPS Positioning software (eps)	
Voice Note-taking software (JustNote)	
Small Arms Ballistics Calculation Software (SBM)	

The programming flexibility of the Zaurus made it the platform of choice with many mobile and wearable computing researchers at MIT, Geogia Tech, and other leading universities. The Zaurus PDA has been used to prototype a wearable health-care platform, next-generation social networking, context-aware sensors and many other projects. While Sharp has been spending millions of dollars in internal context aware and social applications research, it did not leverage or even notice academic research of the highest standard based on their PDA platform.

It is evident that Lead User Innovation Theory (von Hippel, others) is at work in this case. Many of these applications would appeal primarily to technology savvy people or Linux users (a huge market in itself), but a flexible device based on commodity hardware, capable of rapidly adapting functionality based on the users needs, does have mass market appeal. Sharp missed the opportunity to leverage these user innovations and repackage them into next generation products, to the same or different market segments. If Sharp had capitalized on open-source innovation like IBM or a venture-capitalist, and selected ideas that were popular, got the highest votes in the community, or resonated with its corporate customers, perhaps the Zaurus could have done better commercially. It would have been a risk-taking maneuver to release mass market devices around such a radical business model and as a large company perhaps they were not willing to take that risk.

Example 2: Motorola's EZX phones and OpenEZX.org

In mid 2004, Motorola released mobile phones running QT/MontaVista Linux as the operating system, under the EZX platform in Asia and Europe. Most EZX handsets are high-end PDA phones, designed to challenge Nokia's Series 60 Symbian smartphones. In 2005-06, once the the platform was proven, Motorola sold its stake in the Symbian consortium and started switching its commodity phones to Linux, releasing 19 Linux handsets in one financial year.

In terms of the Linux software, there is one principal difference between an EZX phone and Zaurus. The Zaurus is based on an open distribution of Linux. EZX phones use the open-source kernel but a *proprietary* commercial Linux distribution called MontaVista. MontaVista provides phone manufacturers with an off-the-shelf, ready-to-integrate Linux-based phone operating system. However *open-source* Linux developers need proprietary libraries and headers to write useful programs on the phone. In addition, the EZX platform is locked down so that developers and consumers cannot access the Linux operating system directly. Motorola and the service providers believe that by restricting user access they can

create a more secure telecommunications network.

It did not take very long for the open source community to get interested in the EZX platform. Proprietary tools to access the Linux subsystem leaked out, and a very early effort has started at OpenEZX.org. The primary project is to replace the *proprietary* Linux distribution with an optimized, efficient *open* distribution. For example, Motorola's phone-book application is an internal joke with open-source developers, since it has a size of about 6 MB on the filesystem, which is extremely large for what it does.

The OpenEZX group has set ambitious goals. However, most of their current efforts are spent trying to reverse engineer Motorola's platform, and to create a tool-chain and low-level firmware. Motorola internally considers OpenEZX a threat and is trying to enforce crackdown on the leaked firmware flashing tools.

At this juncture, is Motorola making the same mistake that Sharp did with the Zaurus by ignoring the open-source developers? Instead of trying to control or ignore the open source movement, could Motorola leverage fifty-thousand of the world's best and highly motivated programmers for free? Is Motorola willing to acknowledge that, consumers and third-party developers know what applications are best for them, better than phone manufacturers or network operators? Nokia, Motorola's primary competitor is using the middle ground approach by releasing Python (open-source but no low-level access) and the Nokia 770 (completely open source but no GSM stack).

Disruptive Innovation and Profitability

Motorola is currently struggling to understand how to deal with the open source model for the EZX line. There are several ways in which this could play out.

One potential answer might be for Motorola to shift its way of thinking, and consider how it could leverage open source developers with the EZX mobile phones (e.g. make available an

optional open firmware for end consumers). Consumers could decide if they want an open-source phone, and Motorola could potentially reduce its cost of software, akin to Sun Microsystems in the server industry. Nokia has admitted that it would like to be a hardware manufacturer in the future, since software is just an added expense.

Mobile Virtual Network Operators (MVNOs) could be another supporter of open-source. North American MVNOs like Ampd, Helio, and now Disney purchase bandwidth from traditional network operators and retail their niche, branded service to consumers. It is in the interest of MVNOs to support other forms of connectivity because they usually pay the network carrier on a per transaction or megabyte basis. Providing an open-source, highly customizable phone to technology savvy customers might be a significant market segment in itself.

A more radical, yet less likely possibility is a completely open mobile phone stack. There is a conceivable business model around distributing low-cost handsets with open-source software and making profits on the application services, for example, focusing on niche markets like mobile entertainment or dual mode handsets (wifi and GSM).

Acknowledgments:

The author would like to cite the following people as sources of information and influence: Hameed Akhavan (Global CTO, T-Mobile and MIT alum), Timothy Jasionowski (Nokia Ventures), Joe Dvorak and Rohit Chaudhri (Motorola Research), Jamie Hicks (Head, Nokia Lablet, MIT CSAIL), Randy Zadra (Telmex), Jurgen Scheibe (Nokia Champion and Python developer), Andy Lippman (Media Lab Faculty), senior partners at Mayfield Ventures and Highland Capital, Matthew (Nokia 770 open-source developer) and Benjamin Mako (Ubuntu developer). The author is an accomplished hacker on EZX, Series 60 Python and the Sharp Zaurus and would like to acknowledge the efforts of these communities.