

Persistent connectivity for mobile phones

Cheap data's influence on contemporary trends

*Jonas Eriksson (jonas.eriksson@hyperisland.se)
Interaction Design, Master's Program, K3, Malmö University*

Abstract

This short essay aims to provide a broad overview of some of the observable trends arising from recent mobile application development. It will especially focus on the influence on application development of persistent connectivity resulting from recent years advances in speed, coverage and—most importantly—reduced cost for consumers. It does not claim to be a complete and comprehensive overview. Rather, it will focus on a selected excerpt of applications being deployed within the coming months to provide a highly temporal viewpoint; best described simply as *current trends*. The purpose and idea of this essay is to provide inspiration, not documentation.

Introduction

The recent year has, arguably, shown an increased interest in mobile applications from media, developers and consumers and has thus resulted in somewhat of a surge in mobile application development. The reasons are presumably many, and apart from the general incline in deployment of mobile phones since the 80's, could be summarized somewhat according to these three points:

- *Better developer environment*

Mobile application development has traditionally been plagued with restrictive carriers not allowing applications to run on their networks to protect investments or proprietary solutions that could be charged to consumers. Programming languages are many, and the most wide spread such as Java ME and Flash Lite has still not gained any major success. At the same time, multi-device deployment was still expensive, security certificates were associated with cost and there was still no good way to market your application.

Recently this has begun to change. We can see signs of multi-device deployment becoming easier, developer tools becoming better and marketing starting to centralize and carriers opening up their networks. The latter, most prominently to organisations that provide the authorisation process for applications developed within their framework.

- *Better phones*

Devices are becoming more powerful with bigger screens with higher fidelity.

- *Cheaper data*

Data transfers, while still not precisely as cheap as your broadband connection at home, and often limited to in-device use only, have become cheap. While you still can pay a lot per megabyte sent, *unlimited data plans* come with a monthly fee has become affordable, prices in Sweden as of 2008 ranging from 99 SEK for 384kbit/s to around 200 SEK for 7,2mbit/s.

As clear examples of these trends we find Apples iPhone and the Open Handset Alliances (of which Google is most prominent member and often the attributed company) Android with their subsequent bundled carrier deals. While Android is an operating system, the only current Android device available and the iPhone both come with all three points mentioned above; an ecosystem of good developer tools, centralized app stores, large screens, powerful inwards and unlimited data plans.

Method

The intention of this essay is to specifically look at what the latter point, recent emergence of cheap data plans (often bundled or marketed together with relevant devices) has meant for current application development. As stated, the intention is not to be comprehensive, nor is it to predict the future.

The main body of the material comes from the top 50 finalists in the Google Android Developers Challenge [1]. The top 20 of these applications were all as the prize awarded between 100000 – 275000 dollars each. The intention of the essay is thus to look at trends arising from a highly contemporary set of applications launched commercially within the next months. If we accept that mobile application development has recently surged in popularity, this could be seen as looking at emerging trends. First-to-market trends.

Worth noting, concerning the body of the material is that applications were only surveyed via descriptions, video demos and screenshots. No applications are currently available for download, and the first device will not arrive until October. However, time restraints would've forced the analysis method to this either way, as well as the fact that applications cost money – this is also the reason for iPhone applications not being any better material.

Wherever a referred application is not a member of the top 50 finalists, it will be clearly pointed out in the text.

Finally, the trends brought forward are in many cases familiar to interaction designers. My point is not, however, to discover something new. Rather, academic work focused on research has produced a lot of interesting artefacts, design patterns, and scenarios. This essay should simply be seen as an attempt to highlighting some of those as they emerge in contemporary mobile application development today.

Trends

Having established what this short essay will look at, and why, it's time to begin describing the trends. Once again, the point is not to be comprehensive, but rather list the ones that emerge as some of the strongest. Since the material analysed are 50 applications, special attention being paid to the top 20, no single application will be described in full detail but rather relevant descriptions of some will be interwoven in the text where relevant.

In as much as it is possible, I will refrain from discussing the various topics for now and just focus on description. Discussion of these will be saved for the last part of this essay.

Agency

When thinking about agents, it is important to remember that they in this form are inherently mobile. That is, they're portable agents that move about with you in the world. *Pebblebox* [2] acts as a *visible agent*, constantly pulling down information about your surroundings that it deems important to you. *Ecorio* [3], an application helping you reduce your carbon footprint, acts as an *invisible agent* most of the time. It runs as a background process and monitors your travel routes to later, when launched in full, provide you with suggestions on more environmentally friendly ways of travelling the same routes such as carpooling or public transit.

Life360 [4], an application for neighbourhood collaboration to overcome problems together, goes one step further: It allows you to keep up with the safety status of your family, and can initiate alerts for help if it (via the accelerometer) detects that the user is in a car accident.

The *Disco App* (not one of the Android applications) is an application that monitors who of your friends is nearby whom via Bluetooth recognition, and aims to let you know which of your friends are socialising at the moment. The eventual purpose of the concept is to let your mobile phone automatically communicate as much information about your current situation as possible to your friends. Examples apart from the one mentioned above include noisy environment, position, speed, visual setting etc. This information is then used to draw conclusions and augment other contexts, such as "is it worth calling if my friend is at a noisy club?" or "I can call any of these three friends because they seem to be in a car together travelling towards my city".

While Pebblebox does some auto configuration of your phone by switching connection methods to the most energy efficient at the time, *Locale* [5] goes all the way by responding to situations (based on location, time, energy level etc) that you've set up and changing phone parameters such as silent mode, call forwarding, etc to predefined settings.

Seamless (fluency)

Much of storage and processing has moved into the cloud. That is, a central server is sometimes better outfitted to safely store and data. Integration between locally stored data (on your phone) and hosted data becomes a key issue. *Tunewiki* [6] is a media player with seamless integration of locally stored media and media stored online. A video in your "library" that in reality is hosted on YouTube is only distinguishable via a little icon in it's listing, and behaves just as any other song in your library. Tunewiki also pulls down lyrics for the song playing, and synchronises them to the playback, providing online translation of lyrics as the song plays via Google Translate if the user opts to change the language.

Phonebook 2.0 [7] in its essence behaves like a regular contact list, except that it saves and syncs contact information with online servers. Contact profiles can be updated by the actual contact since Phonebook 2.0 provides a social layer on top of the contact list via IM and status messages.

Maps and location

Every other application seems to provide a map interface and/or some sort of location based filtering of geo-tagged information. But some go even further, where location and the map interface is not only the starting point for the application, but in essence also is the application in itself. That is, the location of the user and the map interface is vital to its core functionality and value.

Cab4me [8] exemplifies this. It allows the user to order a cab to his or hers current location with one click. The user doesn't even have to know where they are (specific locations are however available). Choosing destination is also done via dragging the map.

Other applications realize that you might need things that are in the real world, and provide guidance towards it. *Cooking Capsules* [16] is a recipe planning and cooking application that helps you, after having watched a video show of the chosen recipe, to shop for ingredients by showing markets nearby on a map.

Applications like these take their starting point in real world phenomena as opposed to remaining purely virtual (as an application only concerned with showing a map for no particular reason would be). Thus, as expected, map interfaces are often also deployed for showing social information such as friends, feeds of local information etc.

It's so easy to go Social

Many if not most applications allow for sharing in some way of the respective types of data they allow the user to create, or to find. Sharing can be done one-to-many, in such ways as Ecorio that allows you to leave behind inspirational stories of how to reduce your carbon footprint. This essentially becomes one-to-nobody in the sense that the user is not directing the information to anyone specifically.

Sharing is of course also done by sending direct pieces of data to your friends, such as sharing a recipe, or by applications themselves (agency) that provide others with the same application their current location and their activity there. Tunewiki allows users to browse what song are playing where via a map, or where in the world a specific song is being played with Tunewiki right now.

Other applications allow for adding friends and see their activity within the application. *Cooking Capsules* not only allow you to share recipes, display the nearby markets on its map, but will also allow display the position of your friends within that area.

In part, some of the activity specified above points to asynchronous communication becoming a virtue. In cases of one-to-nobody it is often vital since you wouldn't want to disturb, or be disturbed by, a stranger. In cases of one-to-many(-friends), it often begins to embody a peripheral flow of knowledge about ones friends. Phonebook 2.0 enriches the contact list with status messages from friends displayed under their names. Both of these cases could be said indicate some importance of asynchronous communication in combination with passive-awareness.

Adding social factors can also be a tool for making the content of the application even more relevant. Users, especially friends, help each other find the best content, or create it themselves, as we shall see next.

Crowdsourcing

Crowdsourcing is the act of delegating the task of creating or managing content to the users of the application; the crowd. Jeff Howe explains it as

Just as distributed computing projects like UC Berkeley's SETI@home have tapped the unused processing power of millions of individual computers, so distributed labor networks are using the Internet to exploit the spare processing power of millions of human brains. (...) it doesn't matter where the laborers are – they might be down the block, they might be in Indonesia – as long as they are connected to the network. [9]

Geo-tagging information, and often creating most of it, is a perfect example done by many applications.

Pebblebox and Life360 rely on the users to provide most of the content. Life360 helps neighbours post local problems or issues on a map to overcome them together. Pebblebox also allows for geo-blogs and other types of information to be posted to a map and viewed by others.

Breadcrumbz [10] is a navigation tool and route finder that complements the map view with actual in-route geo-tagged pictures and graphics within them, allowing for much higher details such as arrows pointing up the stairs, to the right after the pillar, inside a shopping mall and to the exact door. These routes and images are user created, and the developers encourage users to create routes such as “*with their friends*” (“*find my house*”, “*where I left my car at the airport*”) or *with the world* (“*best mountainbiking route in the area*”, “*find my cool shop*”, “*to your hospital's department of surgery*”, etc.).” [10]

Tunewiki uses crowdsourcing to synchronize lyrics to music. Users simply tap the screen when the current selected lyrics line is sung.

The camera as an eye to the world

Increasingly, the mobile phone can see and feel the world via sensors such as accelerometers, proximity sensors, the microphone, cameras etc: As said previously, Life360 can “feel” a car crash via the accelerometer. While phones have had cameras for several years, the intended usage has still been alike to that of the regular point-and-shoot digital camera; taking still pictures.

Expect the camera to be very important in mobile search, with comparison shopping becoming useful as products similar to the item pictured are found [11]

The camera has become an input device rather than just a recording device. *Compare Everywhere* [12] and *ShopSavvy* [13] both use it to scan product bar codes and provide not only price comparisons and price alerts (including contact info), but rich information about the product such as trailers and reviews.

Google own implementation of their *Street view* service is another example of the mobile phone as a more complex input device. Street view is a display mode of Google Maps where the user is placed on eye level with complete photographic imagery of the surroundings at that position. The application can use the compass to know how the user moves the phone, and subsequently move the scene imagery displayed in a similar fashion. [14] Tracking this type of motion could also easily be accomplished with a camera.

Glanceability

While still nowhere near as “calm” as Ambient Devices [15] artefacts *Ambient Orb*, *Ambient Umbrella* or *Ambient Weather Beacon*, who are often used as examples of the “calm technology” coming out of ubiquitous computing, there is a small trend emerging out of the same concept of *glanceability*. Best defined by Tara Matthews Jodi Forlizzi and Stacie Rohrbach as: “By *glanceable*, we mean enabling users to understand information with low cognitive effort” and “*Glanceability* refers to how quickly and easily the visual design conveys information after the user is paying attention to the display.” [17].

It is easy to see that in the cases where the geographical location of information is so essential to its relevance that it becomes a vital part of the content, maps provides a low effort at-a-glance view.

Mentioned earlier, the Disco App also employs at-a-glance interaction. Since the primary use of the first prototype was to see who was close to whom, users needed not do more than glance at the screen to see whose avatars were grouped.

Discussion

It is now time to discuss the various implications these trends might have. However, I hesitate when using the word *implications*; the characters of these are not definite truths and obviously they’re not recommendations. The idea is not to analyse and discuss from a particular point of view. However, a more open discussion might spur creative thinking in asking oneself; *how can I use this? What can I do with this?* I have therefore opted to no format for the following at all, but allow for both thoughts, questions and, where found, relevant research to be presented.

Agency

Barbara Ballard in *Designing The Mobile User Experience* presents “The Carry Principle” [11] a set of principles to distinguish what she calls “Personal Communication Devices” from regular personal computers. Amongst these, *personal device*, *always on*, *always connected*, *battery powered* and *wireless* have special implications for agency. Your mobile phones knows a lot about who you are, including your position and communication habits. It’s also personal as a device since it is not shared with others. It’s one of the few devices that follow you everywhere you go, and remain switched on and connected in a multitude of social and private contexts.

Maps and location

Location (geographical context) has been thoroughly dissected, discussed and described countless times before – and will not be here. However, what might be worth paying attention to is the fact that in these new ecosystems where a-GPS devices (that is, a GPS device that’s “assisted”, thus the “a”) often come with the phone, users might start to expect location, or maps and geo-tagged information to matter, simply because so many other applications are doing it.

Seamless

Beyond the technical aspects of dealing with (often temporary and short lived) periods of dropped connectivity, leaving the application or interface with some value or functionality is probably desirable. But how to communicate the dropped connection, and when?

What is the access difference between data stored locally and data store in the cloud anyway? It's just a different type of connection. For now, it seems the on favourable difference is the computational (and thus semantic) power available in the cloud.

Social aspects

Is adding a “share” button and showing “friends nearby” mobile application memes? Perhaps—and Ballard might give insight into why when she states:

A communications device is device whose primary purpose is communication. (...) It represents a person's always-available connection to the virtual world, both to information and to people” [11]

Users expect to be able to share and connect with their friends, because it is the essence of the device, and should never be denied that experience.

It could be argued that making this available in even such limited settings or context as recipe and cooking applications is motivated by the fact that it provides valuable context to social connections. If you and your friends really like cooking, why not allow for that social bond within the cooking context. *It is so easy to go social* because the social technology, context and expectations are already present in the device.

Crowdsourcing

Payment aside—because while many crowdsourcing examples on the web do pay contributors, the applications here does not—what makes people create and manage information? The question is probably suitable for another essay, dissertation or book, so I won't try to answer it. However, in the case of Tunewiki, there is a sense that it's both *very easy to contribute* (tap, tap, tap), you might only tribute to *content you like* (*thus enhancing it*), and the result seems to be *visible instantly* (in the same context non the less).

One could also argue that if the content your creating or managing has important ties to the real world, such as documenting locations (your favourite café?), being in the real world with a very competent content creating device is desirable. And if it's a spur-of-the-moment thing, it might often be vital.

Camera

In general, it's interesting to think about how sensors make this personal device see and feel more of the world. In fact, since the first Android phone on the market includes not only a GPS and accelerometer but a compass, it is the first (or at least one of the first few) phones that knows exactly where in the universe it is, how it's positioned and what it's doing.

Concerning specifically the camera, it opens up to think about what we can do with the device when it can begin to, for example, remember the world somewhat as we see it, or make sense of artefacts closer to the way we do rather than via some technical input. We can also think about the possibilities for phone as acting as a controlling input device, monitoring movements through the camera.

Glanceability

Matthews, Forlizzi, Rohrbach has identified four design principles for glanceable displays [17]. Although they seem to deal with standard sized computer monitors, the lessons seem even more important for smaller screens.

(1) Visual elements should *match user expectations*. That is, they should have semantic meaning. If you know the type of information, think hard about what visual can

represent that type. If the type of information is not known; think different. Maybe you do know something about the information, meta-data like when your phone received it, or who it is from and to how many it has been sent, even though you don't know its specific type. (2) *Use abstractions*. Simple icons are better. This is especially important for mobile phones since the screen size is often greatly reduced. (3) Visuals should be *distinct* from each other and from the background. This is even more so true on a smaller display where the display isn't stationary and things tend to get cluttered easily. (4) *Maintain consistency*, and one could argue that it should be done not only within the application, but within the whole ecosystem that is the device.

It could also be good to consider where in your design to draw the line between what is proper at a glance and proper for more interactive use.

Concluding statement

I have tried to describe and present some of the recent trends emerging out of a set of commercial applications all more or less dependent on a persistent or cheap data connection. In part, this could be likened to the contemporary social and environmental analyses that often take place as part of the research phase in commercial projects (i.e. "good examples"). The "why" then is to create an *inspirational* rather than comprehensive documentation, and thus I end with presenting various thoughts that was not appropriate to include in the analysis.

Links and references

- [1] http://code.google.com/android/adc_gallery/
- [2] http://code.google.com/android/adc_gallery/app.html?id=30
- [3] <http://www.ecorio.org/> and http://code.google.com/android/adc_gallery/app.html?id=13
- [4] <http://www.life360.com/> and http://code.google.com/android/adc_gallery/app.html?id=23
- [5] <http://www.androidlocale.com/> and http://code.google.com/android/adc_gallery/app.html?id=25
- [6] <http://tunewiki.com/> and http://code.google.com/android/adc_gallery/app.html?id=48
- [7] <http://www.voxmobili.com/phonebook20/> and http://code.google.com/android/adc_gallery/app.html?id=32
- [8] <http://www.cab4me.com/> and http://code.google.com/android/adc_gallery/app.html?id=5
- [9] The rise of crowdsourcing, Wired, June 2006
- [10] <http://www.bcrumbz.com/> and http://code.google.com/android/adc_gallery/app.html?id=4
- [11] Designing The Mobile User Experience, Ballard, 2007

[12] <http://compare-everywhere.com/> and
http://code.google.com/android/adc_gallery/app.html?id=8

[13] <http://www.biggu.com/> and
http://code.google.com/android/adc_gallery/app.html?id=17

[14] Video: <http://www.youtube.com/watch?v=4PRfVKzuUJ4>
Description: <http://news.cnet.com/google-shows-touchy-feely-android-phone/>

[15] <http://www.ambientdevices.com/>

[16] <http://www.cookingcapsules.com/> and
http://code.google.com/android/adc_gallery/app.html?id=9

[17] Designing Glanceable Peripheral Displays, Matthews, Forlizzi, Rohrbach, 2006