

AppAware: Which Mobile Applications Are Hot?

Andrea Girardello
Information Management
ETH Zurich
Switzerland
+41 44 632 39124
agirardello@ethz.ch

Florian Michahelles
Information Management
ETH Zurich
Switzerland
+41 44 632 7477
fmichahelles@ethz.ch

ABSTRACT

Today most mobile operating systems provide users with an application portal where they can search for applications published by third-party developers. However, finding new apps is not an easy task and requires either to know what to look for or to go through an endless list of applications. In this paper we present work in progress of a platform that allows its users to discover mobile applications in a serendipitous manner. AppAware is a mobile application that captures and shares installations, updates, and removals of Android programs in real time. Accordingly, AppAware allows its users to see what applications are being installed right now or around their position by other people, thus introducing a new way of interaction with application portals and other mobile users.

Categories and Subject Descriptors

H.5.0 [Information Interfaces and Presentation]: General

General Terms

Management, Measurement, Design, Experimentation, Human Factors.

Keywords

Social network, market, mobile, applications, application portal, Android, AppAware.

1. INTRODUCTION

Today mobile phones are everyday companions reaching the attention of research community, commercial industry and consumers. The constant improvement of the hardware related to these devices has enhanced their capabilities, thus letting them run a huge variety of applications, from standalone to client-server ones, and the ubiquitous access to the Internet provides them with an almost infinite information source. In order for these applications to pass from developers to consumers, an established method is that of using an application portal: a developer publishes his application on one of these portals from which the consumer can directly search, download and then install the application onto her mobile device. Application portals are considered nowadays to be an essential element in the distribution process since they play the role of intermediary between developers and consumers, with the potential to reach all

customers from a single virtual shop [1].

The centralized portal, a model in which all applications for a certain platform are published on a single container, has been chosen by two of the today's major mobile phone players, namely Google and Apple. The former is into the development of the Android mobile operating system, which offers a preinstalled application portal called "Market" that allows users to browse, download and install applications published by third-party developers. The Market is divided into categories (i.e. Communication, Entertainment, Shopping, etc.) and for each category three filters are available, namely "Top Paid", "Top Free" and "Just in". The names self-explain their functionalities, but the ranking system for "Top Paid" and "Top Free" apps is not transparent: are applications sorted by number of downloads, active installations and/or average ratings? Moreover, due to the size of the Market that reaches today roughly 40,000 apps, users often have to go through countless titles to find what they are looking for. This entails a prerequisite while searching for a new piece of software on an application portal: the user must exactly know what she needs. As a consequence, making fortunate discoveries of fresh applications by accident might be very limited. With AppAware users learn among each other about meaningful applications based on their installations behavior – similar to sharing of bookmarks, e.g. Delicious¹.

AppAware is a mobile application that shares online installations, updates and removals of Android applications. In this way a user becomes aware (App-Aware) of what other people are installing on their Android phones right now or in her proximity, learning from them. This continuous stream of application events (installations/removals/updates) provides the basis for serendipity for other users. AppAware does not aim at replacing the Android Market, it is rather a companion to discover interesting applications with the help of other users, thus bringing the social factor into application portals (see Figure 1).



Figure 1. Users contribute to AppAware for letting you discover new mobile applications.

¹ <http://delicious.com>

In the following section, we examine the related work in the field of mobile application portals. This is used to deduce some design principle that we describe in section 3, along with the concept. In section 4 we follow up discussing its implications for then summarizing the AppAware idea in section 5.

2. RELATED WORK

In this section, we briefly review the state of the art and related work that have informed our design and indicate how AppAware differs from these.

2.1 Mobile Applications Websites

At present, the Android application portal can be accessed just from the Market mobile application and, in a limited way, from the related website. To overcome these design decisions by Google, many third-party developers are launching new services to access applications' details from a personal computer. These services enable users to search for and download Android applications on the web instead of doing it directly from a mobile device. Good examples are AndroLib² and AppBrain³. The major difference between the two is that AppBrain provides a user with an applications shopping cart that can be synced with the device through an Android client application. However, the idea is not innovative since it is trying to port the concept of Apple's iTunes to the Android world. In fact, iTunes already allows its users to browse and sync applications from their computer to an iPhone. AppAware does not aim at replacing the Android Market or providing a proxy, it is rather a companion to plan users' serendipity [5] in applications finding.

2.2 Appazaar and aTrackDog

Appazaar [2] is a recommender system for mobile applications, and is a project of the Lab for Software Engineering at Münster University of Applied Sciences. Based on a user current and historical locations and applications usage, Appazaar recommends applications that might be of interest for her. Therefore, Appazaar applies different algorithms from the research field of context awareness to analyze all the input data and create profiles of different situations. Another tool related to AppAware is called aTrackDog⁴. It is a program for Android devices that makes sure a user has the latest version of every installed application by checking the release information from either the Android Market, other users' devices or the vendors' web site. Doing this manually takes time, thus aTrackDog supports the user in this activity. Even more, data from users' devices is used to generate a most popular apps list that can be sorted by category, time, and price. Despite AppAware generates similar stats and providing apps recommendations is an appealing feature, we focused towards new ways to explore mobile apps on an application portal (i.e. real time stream, proximity based) and we further underlined the users presence into these activities.

3. CONCEPT AND DESIGN

In this section, we describe the system design, AppAware's most relevant features and their implementation in the user interface.

² <http://www.androlib.com>

³ <http://www.appbrain.com>

⁴ <http://atracdog.a0soft.com>

AppAware shares online users' installations, updates and removals of Android applications. In this way a user becomes conscious of what is hot on the Android application portal. To meet these conditions, the AppAware system consists in a client-server architecture.

3.1 General Concept

The client component in this system is the Android mobile application, which represents AppAware's graphical user interface (GUI) and allows following installations, updates and removals of applications shared by other users. Most of the core functionalities are supported by the main screen (see Figure 2) and are accessible from the application's menu or by touching a list item. Each list item represents a single event with its details, namely: the name of the application with its description, the type of event (installed, updated or removed), the user involved and the Android version together with the phone model. Moreover, to distinguish the type of event at a glance the application's name is colored in red in case of a removal, green for an installation and blue for an update.



Figure 2. Real-time stream of installed, updated and removed applications (a) and analogous events in user's proximity (b).

The server component is a web application accessible through a standard browser and at the same time integrated in the mobile client through an Android WebView element that displays web pages. The client connects to the server through a RESTful interface that accepts and then stores events from users. Among the required parameters we have: the user ID, the application package name (used by the Android Market as unique identifier for an app), event type (installed, updated or removed) and the location (latitude and longitude, whenever allowed by the user). The server offers its data through an RSS feed that the AppAware client uses as data source for its core functionalities as described in Sections 3.2 and 3.3. This design decision allows at the same time any standard feed reader to keep track of installations, updates and removals of Android apps.

Along with this architecture, AppAware integrates with Twitter too. This allows a user to share applications' installations, removals and updates on the Twitter account, thus letting her followers to see what applications are being installed by that user. An example of a Twitter status update is: "Just updated Google Translate <http://appaware.org/1z> on my #Nexus One - via #AppAware". This tweet tells that the application "Google Translate" has been just updated on the Android phone model "Nexus One". The built-in URL loads a web page showing the

history of events and the acceptance rate (see Section 3.6) for the “Google Translate” application.

In the following subsections we comment the new ways of interaction with an application portal introduced by AppAware. We first present four different possibilities to discover novel applications and then how AppAware users implicitly rate them.

3.2 Real Time Stream of Applications

The core feature of AppAware is that of capturing and sharing installations, updates and removals in real time. This creates a stream of events that can be followed from the main screen of the mobile application (see Figure 2a). Here events are ordered by time, with the most recent appearing on top of the list. The list is not refreshing itself automatically, but a small dialog notifies the user about the number of new events since her last refresh. Thus, this screen tells the user what other Android users are installing, updating or removing from their devices right now. This approach increases the likelihood of serendipity by letting a user to accidentally stumble upon a previously unknown application.

3.3 Applications Around You

The second core functionality of AppAware is called “applications around you” and provides a location based service (LBS) tailored for application discovery. In this view the events are ordered by the distance from the current location, thus the user will experience a list of applications installed, updated or removed in her proximity (see Figure 2b). To further increase the turnover of nearby applications, only events within the last week are considered. This LBS is particularly intended for frequent travelers which might be interested in discovering useful applications for a specific context. Examples could be a subway guide for New York City (i.e. NYC Transit, NYCSubway), an application providing information on Switzerland (i.e. Swissinfo.ch) or bus schedules for Madrid (i.e. BusMadrid).



Figure 3. Augmented reality technology used to display applications in the user’s proximity.

A natural extension to this feature is the visualization of applications in the user’s proximity using the augmented reality technology. To achieve this we developed and released a layer for Layar (see Figure 3). Layar⁵ is an application that shows what is around a user by displaying real time digital information on top of the real world as seen through the camera of a mobile phone. Its plugin-based architecture allowed us to quickly explore this field and to provide an additional experience in the applications discovery activity that can be used in parallel with the LBS service integrated in the AppAware client application.

⁵ <http://layar.com>

3.4 Applications Statistics

AppAware provides applications statistics in the form of five lists (see Figure 4a) where applications are ordered with a clear and transparent approach:

- **Top installed:** shows the most installed applications in the last hour, day or week.
- **Top updated:** shows the most updated applications in the last hour, day or week.
- **Top removed:** shows the most removed applications in the last hour, day or week.
- **Top applications for the user’s device:** shows the applications most active for the user’s specific device in the last hour, day or week.
- **Featured applications:** shows the 35 most accepted applications accordingly to the AppAware application acceptance rate (see Figure 4b and Section 3.6).

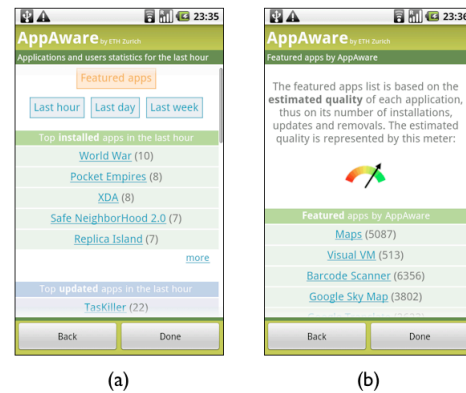


Figure 4. AppAware applications statistics (a) and the list of featured apps (b).

3.5 Users' Applications

In order to allow users to also share recommendations with others on a more personal level, AppAware creates for each user a fingerprint of installations, updates and removals of Android applications (see Figure 5b). This makes comfortable for a user to browse a friend’s applications or even those of any other AppAware users considered trustworthy. A user’s page can be accessed from an application’s event or by searching from the AppAware client for her Twitter username.

3.6 Implicit Applications Ratings

As soon as the AppAware client traces a previously unknown Android application the system creates a web page for it. This page is then accessible from the mobile client by tapping on an application name. On this page the user finds the application’s description, the list of recent users’ events (installations, updates or removals) and the application’s meter representing its acceptance by the community.

At present, several application portals let users rate applications by giving a rating on a one to five stars scale. Despite this being well understood and accurate, it interrupts the usual behavior of a user [3]. Additionally, we suppose that consumers are more likely to review applications if they perceive them as either very good or

very bad, as already found out for movies [4]. To overcome this limitation AppAware presents a new way to let user implicitly rate applications and thus define their acceptance. This acceptance is represented by a meter colored from red to green (see Figure 5a).



Figure 5. An application’s page on AppAware (a) and a user’s fingerprint of events (b).

When the gauge points toward the green range the acceptance is excellent, yellow range for good acceptance and red range if almost no AppAware user is keeping the application installed. The assumption behind this approach is that excellent/good applications are not removed once installed, whereas applications not liked tends to be removed from the device. To model what described so far, AppAware defines the acceptance rate v for an application app as the value going from 0 to 100 computed with the formula in (1), where U is the set of users having at least one event for app .

$$v(app) = \frac{\sum_{user \in U} last(app, user)}{|U|} \quad (1)$$

$$last(app, user) = \begin{cases} 0 & \text{if last event of user for app} = \text{removed} \\ 90 & \text{if last event of user for app} = \text{installed} \\ 100 & \text{if last event of user for app} = \text{updated} \end{cases} \quad (2)$$

As (2) defines, while computing the acceptance rate we consider for each $user$ her most recent event for a certain application app . In this process, an update is considered the highest valuable event, even more important than an installation. The belief is that an application’s update brings to light the user’s uninterrupted interest in that piece of software and, at the same time, the effort of developers in keeping their application up to date.

4. DISCUSSION AND FUTURE WORK

In February 2010 we have freely released AppAware on the Android Market and at the time of writing the response has been faster and more impressive than we could expect. Since the release AppAware counts more than 19,000 unique users, 9,500 of which are active in the last week. At present we have successfully collected more than 1,000,000 installations, updates and removals with an average in the last week of roughly 32,000 events per day. The users have not been asked to try the application not even an advertisement campaign has been run. They voluntarily installed AppAware on their mobile device. Comments from users are very rewarding and AppAware itself seems to be esteemed by the

Android community in general⁶. These readings confirmed us the serendipity effect we were expecting to manifest.

Regarding future work we plan to make an additional step in the user-to-user interaction allowing users to create list of friends and hence follow their installations, updates and removals. Besides enforcing the social factor, another idea keeping us busy is the improvement of the applications acceptance rate model. We could further tune it by considering in the computation the time interval between installations and removals (or days since installation). We can infer that an application keep installed for a long time is slightly better than another application being removed soon after its installation. For doing this our initial assumption that an application is removed if it is not considered good becomes even stronger. However, users might not always remove bad applications, or they could even delete AppAware thus not submitting any removals of previously installed apps. To accommodate this issue, we are considering ruling out inactive users from the computation. We plan also to analyze applications’ average ratings from the Android Market and AppAware applications’ acceptance rates to check whether the two variables are correlated with a certain degree.

5. SUMMARY

This paper has described the current status of AppAware, a mobile application that captures and shares installations, updates, and removals of Android programs in real time. Users can participate to this social market anonymously or share their history via Twitter. By using AppAware, any user can immediately see which applications are hot by watching how many people install or update them. Moreover, users can also see which programs are being downloaded around their location and find the top applications for their specific device. Hence, AppAware introduces several new ways to explore an application portal underlining at the same time the users presence in any of the proposed interactions. Finally, we see great potential in the serendipity effect introduced by AppAware for solving the discoverability problem of today’s application portals.

6. REFERENCES

- [1] Holzer, A. and Ondrus, J. 2009. Trends in Mobile Application Development. In Proceedings of the Mobile Wireless Middleware, Operating Systems, and Applications Workshops.
- [2] Böhmer, M., Prinz, M. and Bauer, G. 2010. Contextualizing Mobile Applications for Context-aware Recommendation. In Adjunct Proceedings of Pervasive 2010 (Helsinki, Finland).
- [3] Claypool, M., Brown, D., Le, P. and Waseda, M. 2001. Inferring user interest. Internet Computing, IEEE, vol.5, no.6, pp.32-39, Nov/Dec 2001.
- [4] Dellarocas, C. and Narayan, R. 2006. What motivates consumers to review a product online? A study of the product-specific antecedents of online movie reviews. In WISE 2006, Evanston, Illinois, USA.
- [5] Eagle, N. 2004. Can serendipity be planned? MIT Sloan Management Review. Fall 2004, vol.46, no.1, pp. 9-15.

⁶ <http://appaware.org/press.php>