World Around Me Client for Windows Phone Devices

Andrey Vdovenko, Anastasya Lukovnikova, Sergey Marchenkov,
Nadezhda Sidorcheva, Stepan Polyakov, Dmitry Korzun
Department of Computer Science, PetrSU
Petrozavodsk, Russia
{vdovenko, lukovnik, marchenk, sidorche, polyakov, dkorzun}@cs.karelia.ru

Abstract

World Around Me (WAM) is a tourist-oriented client application for Windows Phone 7 platform. The primary function is searching in an image hosting website and browsing at the user mobile device favorite photos and pictures associated with the current user location. This demo shows our implementation for HTC HD7 Windows Phone 7 smartphones.

Index Terms: Tourist-oriented client, Geo location, Windows Phone, Flickr.

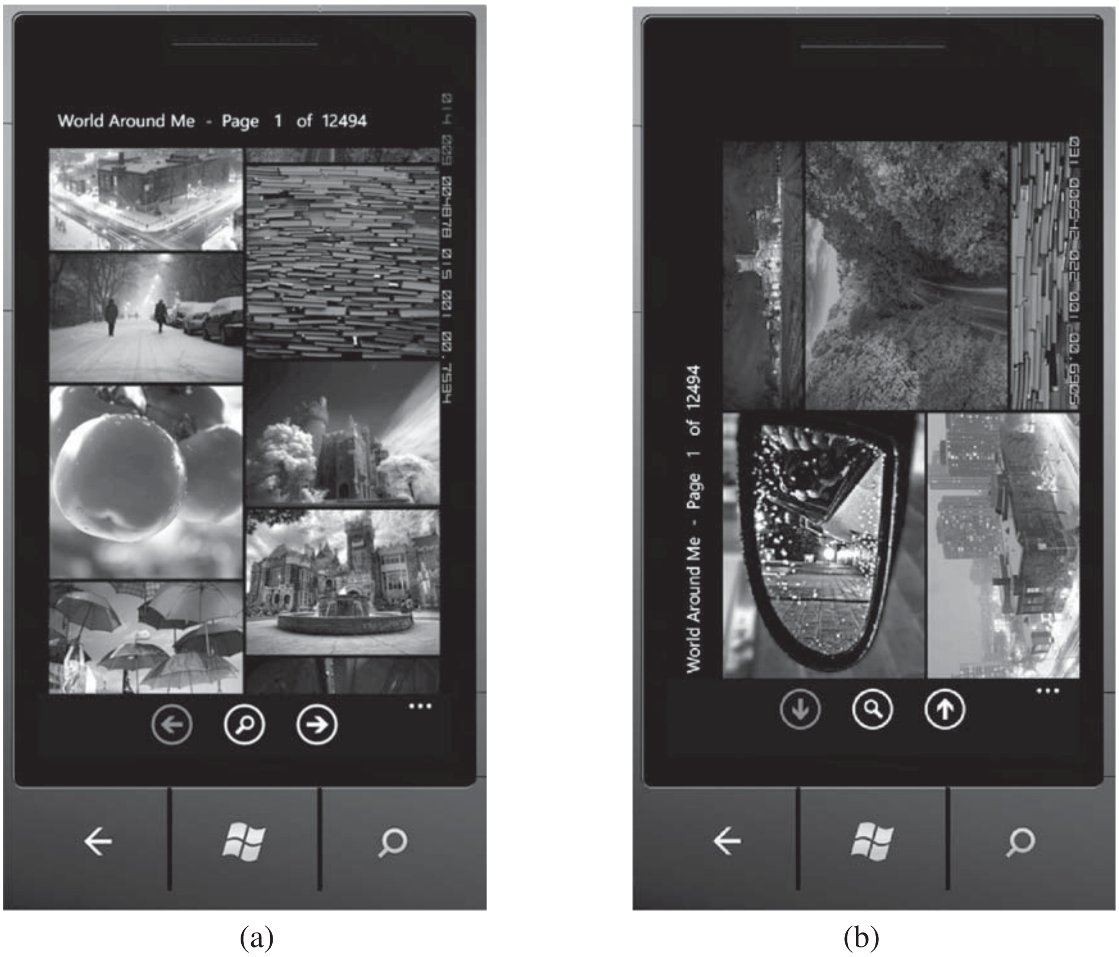
There are huge collections of photos and other pictures collected at image hosting websites such as Flickr [1] and Panoramio [2]. Many people share the media content related to their tourist trips and the places they visited. Also, local residents share content with information about their regional famous places and other points of interest. Further we use term *image* for any graphical object stored at such a website.

Consider the following scenario. A person has arrived to an unfamiliar city (or another geographically localized place). There are a lot of beautiful places around, and she/he has to decide which of them to visit. This decision problem can be quite complicated, especially if the city is big and the time of the visit is limited. In this project, we develop a mobile client application—World Around Me (WAM). It provides an easy solution: with one touch the user browses favorite images of the area where she/he is now.

The area is defined by radius (in kilometers). The user can change the default radius 1.5 km. Now we also elaborate other useful means of the vicinity definition. The user mobile device should be equipped with a geo position system (GPS in our implementation), so WAM can use current coordinates of the user location. As a reference mobile platform and device we use Windows Phone 7 and HTC HD7 smartphone, respectively.

The WAM architecture does not fix a concrete image hosting website. Although to the time of writing this abstract the WAM implementation uses Flickr [1]; drivers for some other websites are in progress. Furthermore, we consider the scenario of multi-service search, when several websites are used as media sources for WAM. In this case, the user selects preferred websites to change the default settings. The multi-service feature will essentially contrast WAM with existing mobile applications of local-media explorers. Let us mention two close applications, Flickr Explorer [3] and GeoFlick [4], which are available in the Microsoft Windows Phone Marketplace application store.

Flickr Explorer supports such advanced features of Flickr as popular picture, random picture and geo location. An essential shortcoming is that the raduis value is static; the user cannot change it. The application does not scale images when browsing, though this feature is crucial for mobile devices with relatively small screens.


World Around Me - Page 1 of 12494
World Around Me - Page 1 of 12494

(a) (b)

Fig. 1. Example screenshots of WAM client running on Windows Phone for portrait (a) and landscape (b) orientation, respectively. Each page displays several images in two columns. The image layout is switched to the WAM original mode when the visualization depends on the image height/width proportions.

GeoFlick provides the same functionality as WAM and also shows on a map the location where the image has been made. The GUI, however, has some defects. It does not use the system application bar, which is a standard solution for Windows Phone applications. Instead, GeoFlick has custom buttons for menu navigation. Another important shortcoming is that during the search process the user cannot access the progress list of found images.

In contrast to Flickr Explorer and GeoFlick, WAM divides the list of found images on pages. Every page is two-column and with 20 pictures per page for effective download and browsing. The example screenshots are shown in Fig. 1. The image arrangement can be made dependent on image size proportions and device screen orientation. To the best of our knowledge, this WAM image layout mode is a unique feature. Optionally, the user can switch on the traditional layout mode when images are identically arranged.

The WAM client allows browsing search results as mini-images to save network bandwidth. Each image is loaded using its link, which the service provided. The result is visualized gradually until the image has been completely loaded; at the same time the user can asynchronously operates with the device. The whole list of links is divided into pages. Page content is downloaded (parallel connections) when the page is active (visible to the user).

Any image can be shown in a separate window, where the image can be scaled using special gestures pinch and tap. Navigation is possible between the images: the common functions *previous* and *next*. An image can be stored in the image gallery (locally on the device) or be sent by email. Information about each image can be depicted as well as the use can see a map where the image was taken.

The client remembers the choice of radius and service the user made the latest time. For the privacy reason, every time when user's current location should be sent to a service, the WAM client asks her/him for permission. The user can input arbitrary coordinates manually, using the menu. In future we also plan to implement an advanced scenario when the user types a name of the place, receives its coordinates and then browses related images.

The WAM code is divided into GUI and Logic, see Fig. 2. This code structure simplifies the development. Since GUI does not depend on Logic we can modify them without causing major changes in the opposite side. For now we use only Flickr service; in near future we plan to add Panoramio service.

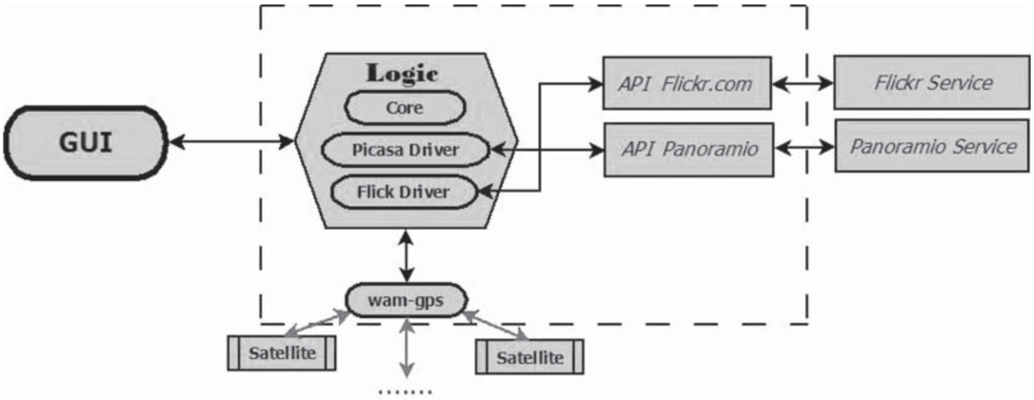


Fig. 2. The WAM arhitecture: separation of GUI specifics from the client logic.

The client logic is written in .NET Compact Framework using Visual C#. Logic architecture consist from Core, Drivers for services and GPS module. This architecture provides the extensibility in adding new services (websites) and functionality. The Core redirects requests from GUI to appropriate drivers. They forward request to associated services and receive links to images from them. Using these links, WAM can dynamically download images and show them to the user. The GPS module determines current geographical coordinates and passes them to the Core.

WAM GUI is implemented with Silverlight for Windows Phone [5] using Visual C#. The application design separates graphical representation from client logic. The GUI design style follows the Metro UI rules, see Fig. 1 for illustration.

REFERENCES

[1] "Flickr: photo sharing website," Feb. 2012. [Online]. Available: http://www.flickr.com/

[2] "Panoramio: Photos of the world," 2012. [Online]. Available: http://www.panoramio.com/

[3] "Flickr Explorer," Feb. 2012. [Online]. Available: http://www.windowsphone.com/en-US/apps/e4d096c4-4216-42d8-ab01-951320db8c34

[4] "GeoFlick," Feb. 2012. [Online]. Available: http://www.windowsphone.com/en-US/apps/834ba53d-8768-e011-81d2-78e7d1fa76f8

[5] C. Petzold, *Programming Windows Phone 7*. Microsoft Press, 2010.