Contents lists available at ScienceDirect
Astronomy and Computing
journal homepage: www.elsevier.com/locate/ascom

Full length article

Mobile applications and Virtual Observatory

A. Schaaff[a,b,*], S. Jagade[c]

[a] Observatoire astronomique de Strasbourg, Université de Strasbourg, CNRS, UMR 7550, France
[b] 11, rue de l'Université, F-67000 Strasbourg, France
[c] IUCAA, Post Bag 4 Ganeshkhind, Pune University Campus, Pune 411 007, India

ARTICLE INFO

Article history:
Received 2 November 2014
Received in revised form
10 March 2015
Accepted 11 March 2015
Available online xxxx

Keywords:
Mobile application
Virtual Observatory
Astronomy
Smartphones
Internet tablets

ABSTRACT

Within a few years, smartphones and Internet tablets have become the devices to access Web or standalone applications from everywhere, with a rapid development of the bandwidth of the mobile networks (e.g. 4G). Internet tablets are used to take notes during meetings or conferences, to read scientific papers in public transportation, etc. A smartphone is for example a way to have your data in the pocket or to control, from everywhere, the progress of a heavy workflow process. These mobile devices have enough powerful hardware to run more and more complex applications for many use cases. In the field of astronomy it is possible to use these tools to access data via a simple browser, but also to develop native applications reusing libraries (written in Java for Android or Objective-C/Swift for iOS) developed for desktops/laptops. We describe the experiments conducted in this domain, at CDS and IUCAA, considering a mobile application as a native application as well as a Web application.

1. Introduction

Some members of the IVOA[1] (International Virtual Observatory Alliance), developing mobile applications since several years, have now skills in native iOS and Android developments, Web development (especially HTML5, JavaScript, CSS3[2]) and conversion tools (e.g. PhoneGap[3]). Development in HTML5/JavaScript/CSS3 is promising and it has the advantage of allowing developments independent from the evolution of the mobile platforms (''write once, run everywhere'').

Our approach was to perform testing of hardware, to develop prototypes but also operational applications or services. Compared to traditional developments one of the biggest change comes from human/computer interaction that is radically modified by the multitouch use. This interaction requires a redesign of the interfaces to take advantage of new features (simultaneous selections in different parts of the screen, etc.). In the case of native applications, the distribution is usually done through on-line stores (App Store, Google Play), which give a visibility to a wider audience. As the frontier is becoming thin between a native application and a Web application we use mobile application as a generic expression in this paper. To simplify, we use also desktop to designate both desktops and laptops. In this paper we give an overview of the experiments we have conducted and the applications we have developed. Mobile devices are not restricted to Android and iOS but we have focused our efforts on the most popular. For those wishing to achieve this kind of development, we give also technical information (development frameworks, languages, etc.).

2. Mobile application for the Virtual Observatory

2.1. Motivation

The IVOA does not develop specific standards for mobile devices, its aim is to work on interoperability of applications and services based on the definition of standard data models, data access layers, registries, etc. Mobile devices and their applications benefit from these standards and also from previous developments. It is for example possible to reuse libraries, developed for desktops, in mobile applications. The high improvements in the network bandwidth now allows to use a smartphone to launch jobs on servers, to store or move data on clouds or on large dedicated storage spaces like VOSpace (Graham et al., 2011). VOSpace is the IVOA interface to distributed storage. It specifies how VO agents and applications

* Corresponding author at: Observatoire astronomique de Strasbourg, Université de Strasbourg, CNRS, UMR 7550, France.
E-mail address: andre.schaaff@astro.unistra.fr (A. Schaaff).

1 http://www.ivoa.net/.
2 Cascading Style Sheet version 3.
3 http://www.phonegap.com/.

http://dx.doi.org/10.1016/j.ascom.2015.03.006

can use network attached data stores to persist and exchange data in a standard way.

Concerning the Web developments, it is now necessary to take into account the diversity of devices to implement a responsive Web design (the user interface changes automatically following the device display characteristics). The possibilities are also broader with a mobile device because it brings new features. It is for example possible to use gyroscope, accelerometer and GPS (Global Positioning System) to point an object in the sky. The IVOA has an Interest Group dedicated to Education. This domain should also be a good experimentation field for mobile applications. When possible, a VO service should be accessible from everywhere and on every device.

2.2. The use cases

CDS and IUCAA have not joined forces for developments, but have decided to share their experiences.

IUCAA was interested by the development of mobile applications mainly for Education (for example Stat-Lite) and its experiments were based only on Android. This choice is motivated by the percentage of people with an Android device and the effort at the Ministry level to furnish Android Internet tablets.

CDS was interested by the R&D aspects to learn how to develop mobile applications in several ways and to find use cases in connection with its services and tools. The survey visualization application was based on HEALPix[4] which is now widely used at CDS. The proof of concept concerning the use of an Internet tablet as a remote control is now used in the frame of a European project, for its outreach.

We give more details directly in the concerned sections.

3. How to develop a mobile application?

We have explored three ways to build applications for mobile devices.

A first way is to develop a Web application in HTML5/JavaScript/CSS3 like for desktops and with dedicated JavaScript libraries like jQuery[5] Mobile which are optimized for mobile devices. jQuery Mobile is a HTML5-based user interface system designed to make responsive Web sites (the user interface is adapted following the device display characteristics) for smartphone, tablet and desktop devices. At CDS, a significant part of the developments is based on these technologies and the developers have de facto the skills to learn quickly how to develop applications for mobile devices through this way.

A second way is to develop a native application in Java for Android and Objective-C/Swift for iOS. In astronomy it is now relatively common to develop in Java but probably not in Objective-C/Swift. For this reason, in the second case the maintenance and the evolution will be more random. A native development could be motivated by the re-use of existing libraries to gain time rather than re-develop in an other language, above all if it implies a scientific expertise. This is often the case in astronomy.

The last way is to develop Web applications in HTML5/JavaScript/CSS3 and to convert them in pseudo native code through converters like PhoneGap. It is not a true conversion in Java or Objective-C/Swift but an execution in a Virtual Machine.

We have not experimented an other solution based on the conversion from a language (for example Java) to an other one like Objective-C/Swift. A few frameworks do it like J2ObjC[6] which converts just the Java code (data access, etc.) but not the GUI side which must be still developed in the native language.

4 http://healpix.jpl.nasa.gov/.
5 http://jquerymobile.com/.
6 https://code.google.com/p/j2objc/.

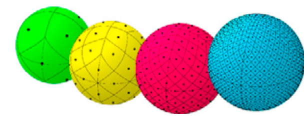

Fig. 1. HEALPix principle.

4. Experiments and developments at CDS

4.1. Introduction

The CDS started to develop for mobile devices in early 2010 (Schaaff et al., 2013). We give here on overview of the most significant experiments and developments done in the frame of R&D (Schaaff et al., 2014), more information is available on the CDS Developer's corner.[7] We often refer to Aladin Sky Atlas,[8] a portal of the Virtual Observatory, developed at CDS.

4.2. HEALPix viewer

HEALPix[9] (Hierarchical Equal Area isoLatitude Pixelization) offers a way to display large image surveys through a pixelization of the sky (cf. Fig. 1 for the principle). The number of HEALPix (Górski et al., 2005) surveys is growing and a few HEALPix visualizer are available like Aladin through the Allsky mode (Fernique et al., 2010). This mode runs well on a desktop with good performances and it was a real challenge to develop a HEALPix visualizer for smartphones and Internet tablets in 2011.

We decided to develop it (Fig. 2) on Android to reuse a part of Aladin Java code. We used Eclipse[10] and ADT[11] (Android Developer Tools) plugin for Eclipse to develop this application. A tablet linked to the development platform was mandatory as the emulation device was unable to execute the application. The emulation device worked fine only for small developments requiring few resources.

First tests were not really good due to the hardware capabilities (in 2011) compared to desktops. With 1.5 FPS[12] it was not usable. For a good user experience the number of FPS should be at least 30.

Aladin was not using OpenGL[13] (Open Graphics Library) but this graphical library was available on almost all the mobile devices. OpenGL is a cross-language, multi-platform application programming interface (API) for rendering 2D and 3D vector graphics. The new prototype, called SkySurveys (Schaaff et al., 2013), implementing this library was between thirty and forty times faster with 45 to 60 FPS. It was sufficient to provide a fluid multitouch interaction.

All the surveys available and added in the future in Aladin will also be available in SkySurveys. We consider this application as an advanced prototype for Android devices. We will not develop the same application for iOS devices because the porting of Aladin HEALPix Java libraries needs resources to do it and to test. It involves also OpenGL which is not trivial to manipulate. The maintenance of an application for both Android and iOS has also a cost we cannot assume. A solution could be to switch from OpenGL to WebGL[14] and thus to develop with HTML5/Javascript/CSS3. An effort is on going concerning the visualization with HTML5 canvas.

7 http://cds.u-strasbg.fr/resources/doku.php.
8 http://aladin.u-strasbg.fr/.
9 http://healpix.jpl.nasa.gov/.
10 https://eclipse.org/.
11 http://healpix.jpl.nasa.gov/.
12 Frames per second.
13 https://www.opengl.org/.
14 https://www.khronos.org/webgl/.

Please cite this article in press as: Schaaff, A., Jagade, S., Mobile applications and Virtual Observatory. Astronomy and Computing (2015), http://dx.doi.org/10.1016/j.ascom.2015.03.006