

Challenges for Mobile Gaming based on AJAX

Dieter Van de Walle, Nico Goeminne, Frank Gielen, and Rik Van de Walle

Ghent University – IBBT

Department of Electronics and Information Systems - Multimedia Lab

Department of Information Technology - IBCN

Gaston Crommenlaan 8/201, B-9050 Ledeborg-Ghent, Belgium

{dieter.vandewalle,nico.goeminne,frank.gielen,rik.vandewalle}@ugent.be

1. Introduction

1.1. Mobix goals and activities

The authors represent the start-up company called 'Mobix Technologies'. This company is a spin-off company of the University of Ghent (Belgium) in cooperation with the Interdisciplinary institute for BroadBand Technology (IBBT, [1]), a research institute founded by the Flemish Government. The IBBT is a sponsoring member of the World Wide Web Consortium.

Mobix tries to overcome the chicken-and-egg problem internet on mobile devices is currently faced with: content providers do not invest in mobile content because there is only a small userbase for it. However, the lack of content that is accessible through mobile internet is certainly one of the reasons for the limited adoption of mobile internet.

We are developing a technology that has the potential of breaking this deadlock. Our technology enables a semi-automatic translation of 'normal' webcontent (aimed at desktop devices) in such a way that the adapted content is suited for mobile devices. This adaptation takes into account the properties of the target device, the semantic properties of the content being adapted as well as the mobile user's location context and personal preferences. Every mobile user receives a personal version of the web on his phone. Our adaptation process is semi-automatic by purpose: it allows website developers to optimize the user experience while still relying on automatic translation for the major part of the content. As a result, mobile websites can be built at lower cost and with better personalized user experience.

We are also active in the development of browser-based mobile gaming applications. AJAX is one of the technologies we use to enable interactivity in these applications. In this paper we will share our experiences using Mobile AJAX. We will highlight some problems we encountered and discuss possible solutions.

1.2. Motivation for attendance

We expect mobile content will go through the same evolution that content aimed for desktop devices has gone through: an increasing level of interactivity of the content. AJAX will certainly play a big role in enabling interactivity in mobile content. Therefore it is important for us to keep in touch with the ongoing development and future evolutions in this area. The workshop on Mobile AJAX offers a unique opportunity to get an insight into the future of Mobile AJAX. We welcome the opportunity to attend the workshop and will gladly share our experiences and views on Mobile AJAX and mobile internet.

2. Mobile gaming

2.1. The Mobix gaming platform

The best way to adopt new technology is by playing with it. Mobix supports this approach by providing a multiplayer online mobile game platform. Brand owners, sponsors, or operators can easily create multiplayer games that are easy to play and that expose the user to mobile internet usage. The Mobix game platform stimulates the usage of familiar phone functions (e.g. SMS, camera, location module) to immerse the player into the game. The platform is designed in such a way that it supports “user generated games” where users can input their own game format and game play on the desktop game builder and release it to the public.

2.2. Mobile limitations

2.2.1. AJAX vs. Mobile AJAX

Mobile AJAX is not just regular AJAX on a smaller screen. There are numerous factors that differentiate Mobile AJAX from its desktop counterpart, the obvious issue being the form factor of mobile devices. They are small and as a consequence bear limited input and output mechanisms.

Some of the limitations Mobile AJAX is faced with are very similar to the limitations AJAX on desktop devices was faced with in its early days. The building blocks of AJAX were already available in 1997 (DHTML and Microsoft’s XML over HTTP control). But time was not yet ready for the technology to be generally adopted. At that time, the limited capacity of the internet connections, limited computing power, and lacking general support for the AJAX key technologies did not make AJAX an attractive option. However, times have changed: broadband internet connections have become the standard, Moore’s law held true, and support for the AJAX technologies has become better.

Considering mobile devices, it is easy to spot the analogy. The range of mobile devices in use today is extremely diverse, but in general we can say the same limitations apply to the greater part of current on-market devices. The available bandwidth for a mobile device varies greatly depending on the network type used, but most commonly used network types nowadays can safely be compared to the dial-up connections available in 1997. The same roughly applies to the computing power. It seems AJAX support for mobile phones marketed today is becoming standard, but looking at the phones currently in use we see full AJAX support cannot be taken for granted. However, looking back at history, we can be fairly confident these restraints will be solved.

Consider the internet connection available on mobile devices. It is a reasonable assumption that broadband internet access for mobile devices will become widespread at some point in the future. However, the technology used to realize this wireless broadband access does not stand comparison to wired broadband access. Latency times are typically a lot less favorable for mobile connections and connection quality can vary. There is not even a guarantee that a connection will be available at all.

Another issue that should be kept in mind is that mobile devices are battery powered. Javascript/EMCAscript [2] execution drains large amounts of power from the battery, and the same applies to sending and receiving wireless data.

And while most wired broadband connections use the flat-free pricing structure, this is still rather uncommon for mobile device data communication. Intensive, prolonged usage of the mobile data service often results in expensive data fees.

If an acceptable user experience is to be offered, the above factors have to be taken into account while developing AJAX applications, and effort should be made to minimize the discouraging effect of these factors in the future.

2.3. Gaming experiences

While developing our game platform, we have run into several limitations that need to be addressed. We try to make the games as immersive as possible by involving many multimedia elements. A typical example of a multimedia feature that is commonly available on a mobile device is a camera with the possibility to take pictures and to make videoclips.

In one of our concept games, at a certain point it is necessary for the player to take a picture and post it to our server. However, this proves to be not so trivial for most users. The user needs to exit the browser application and navigate to the camera function. After taking the picture (with the right camera setting!) he or she needs to return to the active browser window and select the control where the file can be uploaded (if the browser has not already been closed by selecting the camera feature). The user has to navigate to the right location in the filesystem of his mobile device and select the file which holds the most recently taken picture. After activating the 'upload'-button the file gets transferred to the server and the transaction is finished.

This process is rather lengthy. Several steps have to be taken in order to finally send the picture to the server. The input mechanism of the mobile device makes this process extra tedious. It also poses several challenges to the user. A good knowledge of the usage of his or her mobile device is expected from the user.

Another feature that offers significant possibilities for mobile gaming is the location module, integrated in many high-end devices nowadays. However, taking advantage of this feature from an AJAX application is next to impossible. It is possible to work around this problem, but not without using a helper J2ME application. Of course, this is far from desirable. J2ME clearly has the advantage when it comes to interfacing with the device. It is striking we have to resort to using J2ME in order to access this functionality.

In order to provide a streamlined gaming experience, these limitations should be overcome. In general, we notice that there is a need for a closer relationship between AJAX and the device-specific features. The inability to access device-specific features like the camera, the filesystem, the dialer, and/or the location module poses a severe limitation on AJAX applications. Therefore we propose that a general, standardised API should be available for AJAX applications that offers the possibility to interact with the device-specific features and services available on the host device.

3. Access to device-specific features

Content providers currently have two options for implementing rich mobile applications: AJAX or J2ME. Both have their advantages and disadvantages.

3.1. AJAX vs. J2ME

J2ME offers API's that allow accessing device-specific features, and more API's are being developed through the Java Community Process Program [5]-[7]. J2ME support is available in some form on almost any mobile device on the market right now. However, the user has to download the application to his or her device before it can be used, which negatively influences the deployment of the application. In spite of the standardized API's offered by the J2ME platform, application development is far from "write once, run everywhere". The application needs to be adapted for every subset of mobile devices available, which makes developing J2ME applications expensive and time-consuming.

AJAX does not suffer from these problems. The browser is responsible for providing the abstraction to the device, this way far less effort has to be put into making the applications compatible with the target devices. AJAX applications are also easier accessible. No download is required from the user. Accessing an AJAX application is as easy as clicking a link. However, shifting responsibility to the browser also brings certain disadvantages. AJAX is heavily dependent on the browsing agent being used, and the conformance of the browsing agent to the relevant standards. The abstraction offered by the browsing agent is contradictory with the need for access to device-specific features. No mechanism is available to take advantage of these features.

Both platforms have their strong points. J2ME is aimed at feature-richness and performance, while AJAX favors device-independence and easy access by a wide audience. However, the device-independence of AJAX is both a blessing and a curse. The inability to access device-specific features imposes a severe limitation on the use of AJAX technology. Content providers requiring access to such device-specific features are currently forced to use the J2ME platform, in spite of the porting problems and higher accessibility threshold. If a standardized, cross-platform, well-supported solution would be available for accessing device-specific features, AJAX would adopt many of the benefits of the J2ME platform, while still retaining its advantages over J2ME.

3.2. Current solutions

We are not the first to address this issue. Solutions have already been proposed. We will discuss two solutions that are known to us.

3.2.1. Opera

The Opera webbrowser supports some proprietary Javascript API's that enable access to the device-specific features of the host. We were unable to find much information about this feature, but we did however find information about an API supporting access to the remote of Nintendo's Wii console [3].

While this kind of solution offers the functionality we desire, this is a proprietary, non-standardized solution and requires the presence of a version of the Opera browser supporting these API's. We fear this may lead to fragmentation and cause confusion among users.

3.2.2. Open Mobile Alliance

The Open Mobile Alliance (OMA, [8]) has released the OMA External Functionality Interface V1.1 specification [9] which is currently in the Candidate Enabler fase. This specification fills the needs we have adressed above: an open, standardized architecture for accessing device-specific services from an application running inside a user agent (the browser). The specification is marked as optional in the OMA Browsing specification [10].

The OMA External Functionality Interface has the potential to provide a solution for the concerns we mentioned. We strongly hope the OMA EFI specification will move into the Enabler Release fase in the near future, and that browsers and mobile device manufacturers will pick up the specification and provide support for it.

4. Conclusions

It is expected that usage of mobile internet will take a leap somewhere in the near future. Mobile web standards are evolving into maturity and Mobile AJAX is one of the technologies that offer promising opportunities. A major competitor is the Java J2ME platform. While each technology has its advantages and disadvantages, we feel Mobile AJAX suffers from its inability to take advantage of the device-specific features of the mobile device. Some solutions have already been proposed, but at the moment there is no open, well-supported, standardized solution available to content providers.

We would like to emphasize the importance of a standardized, open solution to this problem. If such a solution is available and well-supported, it can allow Mobile AJAX to gain the advantage over the competing J2ME platform.

5. Acknowledgements

The described research activities were funded by Ghent University, the Interdisciplinary Institute for Broadband Technology (IBBT), the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT), the Fund for Scientific Research-Flanders (FWO-Flanders), the Belgian Federal Science Policy Office (BFSPO), and the European Union.

6. References

- [1] <http://www.ibbt.be/>
- [2] <http://www.ecma-international.org/publications/standards/Ecma-262.htm>
- [3] <http://dev.opera.com/articles/view/the-wii-remote-api/>
- [4] <http://jcp.org/en/home/index>
- [5] JSR-179: Location API
- [6] JSR-135: Mobile Media API
- [7] JSR-234: Advanced Multimedia Supplements
- [8] <http://www.openmobilealliance.org>
- [9] http://www.openmobilealliance.org/release_program/efi_v11.html
- [10] http://www.openmobilealliance.org/release_program/browsing_archive.html