Issues for mobile Ajax for cellular users

Goro KUNITO
Research Laboratories, NTT DoCoMo, Inc.
3-5, Hikari-no-oka, Yokosuka, Kanagawa, 239-0807 JAPAN
kunitou@nttdocomo.co.jp

Abstract

Unlike fixed network environment, mobile network environment has several problems to provide web services. This paper analyzes the differences between these environments, and presents issues of Ajax development for mobile browsers.

1 Introduction

As the number of mobile users increases, so does the need for Web access from mobile handheld devices. The current disparity between such devices' available resources and the resources required for smooth Web browsing makes it difficult and unpleasant to access Web pages. The mobile network has narrow bandwidth and disconnects frequently, and the mobile devices has limited power supply and user interface.

In the following sections, we make a survey on Japanese mobile phone market first, and describe issues to develop mobile browsers with Ajax.

2 Japanese mobile phone market

The number of mobile phone subscriber is 98 million, and NTT DoCoMo Inc. has a 53.7% share. Although the number of fixed broadband Internet users is 46 million, the number of mobile Internet user is 85 million. More people tend to access the Internet through the mobile network rather than the fixed network in Japan.

Japanese users send Email instead of SMS and use the pictograms in the Email. Recently decorated mails, which has colorful, blinking characters and attached image, are becoming popular especially among young users. In fact the decorated mail is a HTML mail and PC users can also read it with a web browser. Almost all mobile phones have a camera and pictures are sent by Emails. Smart-phones are not very popular and only few people and business persons carry a laptop PC.

It is said that Japanese mobile Internet succeeded most in the world and the key to this success is adopting Compact HTML instead of WAP since contents provider was more familiar with CHTML than WAP Although the capability of a web browser on a mobile phone is very low than that of PC now, if the capability of a web browser on a mobile phone catches up with that of a PC, a mobile phone can be expected to rival a PC as an Internet terminal.

3 Issues

3.1 Narrow bandwidth

Program scripts written by AJAX have to be downloaded from server before providing the services, and user data stored on the servers also have to be downloaded as the need arises. For an example, the size of main.js of Google Maps[1] is more than 200KB, and the size of downloaded data is more than 180KB when I access my GMail[2] account. Since current maximum downlink bandwidth of 3G system is 384Kbps[1], one has to wait for several seconds when he starts the services over the 3G network. Compared to the conventional web pages, quick response is one of the advantages of web sites with AJAX. Therefore it is im-

[1] the maximum downlink bandwidth of 3.5G system is 3.6Mbps

portant to reduce the latency of downloading scripts. Several solutions can be considered as follows;

- to cache scripts on the mobile phone
- to divide scripts into small subsets in order to download the minimum and necessary subsets
- to download scripts dynamically and asynchronously

3.2 Frequent disconnection

In order to support web services in offline environment, several projects propose or provide offline storage functions, such as GoogleGears[3], DojoOffline[4], DOM Storage[5] and Offline Resources[6].

We give a simple use case using a web services.

> An user uses a web service with his PC connected to his office network in the office. When the user is out of the office, he uses the offline.

In this case, since the state of the connectivity of the PC is obvious to the user, it is easy to switch modes of the service between offline and online manually. Offline storage functions described above can be expected to work well under this environment.

On the other hand, when the user uses the service under the wireless communication environment, wireless link is often disconnected when the user moves around. The disconnection occurs so unpredictably that users can not change mode ahead. In order to make full use of online connection period, it is important to detect wireless reconnection rapidly and to perform some processes, such as synchronization, to be done immediately after reconnecting. To achieve this, browsers on mobile terminal needs to poll network condition or need to be triggered by the change of connectivity status. Mobile carriers may have chances of contribution to detect condition of wireless network to improve performance.

3.3 Limited power resources

The capacity of a battery on a mobile phone is limited because of weight and volume. A current typical mobile phone loads a Lithium-ion battery whose capacity is about 700mAh, and average talk time and standby time are up to about 3 hours, 25 days respectively. Recently many web pages contain scripts, which require more power to render than static HTML. We are concerned that many web designers use scripts even if they make a simple web page which can be created without any script. We believe that scripts should be used as less as possible.

3.4 Limited user interface

Compared to desktop PCs, since a mobile phone has a small and low resolution display, browsers have to adjust web pages according to the display. As well as output interface, input interface has limitation. Since most mobile phone have a ten key instead of a full key keyboard, and multiple characters are assigned to one key, it is important to reduce the number of keystrokes. To achieve this, shortcut and key press event adjusted to ten key may be useful.

4 Conclusion

This paper analyzes the differences between fixed network environment and mobile network environments, and presents issues to develop mobile browsers. As described above, problems still remains to be solved in order to use JavaScript / Ajax in mobile web services. However, since superior user experience provided by Ajax is important for mobile network users as well as fixed network users, we would like to discuss these issues.

References

[1] "Google Maps." `http://maps.google.com/`.

[2] "Gmail." `http://mail.google.com/`.

[3] "Google Gears: Enableing Offline Web Applications." `http://gears.google.com/`.

[4] "The Dojo Offline Toolkit." `http://dojotoolkit.org/offline`.

[5] "Client-side session and persistent storage of name/value pairs." `http://www.whatwg.org/specs/web-apps/current-work/multipage/section-storage.html#storage`.

[6] "Marking Resources for Offline Use." `http://www.campd.org/stuff/Offline\%20Cache.html`.