Intelligent Mobile Web Pre-fetching (IMWeP) Using XML Technology

Sarina Sulaiman, Siti Mariyam Shamsuddin
Soft Computing Research Group
K-Economy Research Alliance
Universiti Teknologi Malaysia
81310 Johor, Malaysia.
sarina@utm.my, mariyam@utm.my

Ajith Abraham
Machine Intelligence Research Labs (MIR Labs), Washington 98092, USA.
http://www.mirlabs.org
ajith.abraham@ieee.org

Shahida Sulaiman
School of Computer Sciences
Universiti Sains Malaysia
11800 USM, Penang, Malaysia.
shahida@cs.usm.my

***Abstract*—Both Web caching and pre-fetching is a ubiquitous technique that can increase the speed of Web loading process. As mobile context has limited resources like speed and memory, a better technique for Web loading process is vital. This paper proposes a new technique known as an Intelligent Mobile Web Pre-fetching (IMWeP) that creates pre-fetching in a mobile environment using the Extensible Markup Language (XML) technology. This technique proves to be faster in searching or accessing process in mobile applications. A case study on Facebook Mobile investigates the loading process by creating priority ranking on XML files using the proposed IMWeP technique. The result shows that the proposed technique provides faster loading process as the result of the incorporated pre-fetching technique and the tree-view method in the proposed system.**

Keywords-Intelligent Mobile Web Pre-fetching; XML; mobile environment; Facebook Mobile

I. INTRODUCTION

Pre-fetching technique can reduce bandwidth load and user-perceived latency. Furthermore, pre-fetching is suitable for traffic shaping and idle-time processing [1][2]. There are a lot of elements that contribute in reducing Web speed such as heterogeneous network connectivity and the distance between server and client. Web caching is a technique that can trim down the network and the server load. By injecting Web pre-fetching into local caches, it can act as an agent to decrease the latency [3][4][5].

Web pre-fetching is an interesting technique that can be used to reduce network resource demand. Furthermore, the combination of pre-fetching and Extensible Markup Language (XML) technologies will give a momentous enhancement of Web infrastructure mainly in a mobile environment. A mobile environment has limited screen size, memory, and the speed of the process. Therefore, a number of techniques and models have been proposed to handle those limitations.

We believe the pre-fetching problem can be solved in two steps. The first step is to search the word that matches with URL and then write it to the XML file. In our study, this locality is estimated with high access probability that will be requested by the users very soon. When the access probability of an object is achieved, the next issue is on how to determine the efficient way to fetch the file. Pre-fetching on a mobile environment can be created by using the technology such as .NET compact framework and J2ME. The second step is to read the associate file according to username and password that has been entered in the first stage. Thus we propose an Intelligent Mobile Web Pre-fetching (IMWeP) that creates pre-fetching in a mobile environment using XML technology. Our study applies HTC Touch 3G Personal Digital Assistance (PDA) model to implement the proposed technique.

The following Section 2 presents the related work on Web caching and pre-fetching that explains an existing work related to a mobile environment. Section 3 presents the proposed technique and Section 4 describes the evaluation on IMWeP using Facebook Mobile as the case study. Finally, we discuss the findings and conclude the study in Section 5.

II. RELATED WORKS

A number of researches investigate the way to improve current Web caching and pre-fetching techniques. Said *et al.* [6] create a reserve data before the data loaded by a user. This data is prepared by a pre-fetching manager by storing the request of user in a server, and it will be organized according to users' request. The related data will be sent back to the client either using a smart phone or a PDA and the non-related data will be ignored. The proposed method in the work is proven can increase and handle the limitation of a mobile device.

Hong *et al.* [7] suggested a different way on pre-fetching by creating transcoder based on XML technologies that converts the stream of multimedia content in the XML file. This will allow unnecessary data on stream to be removed and only related data in a stream will be transferred to a mobile device of a client. The transcoder has a capability to set up the data to be suited with bandwidth of the mobile network as well.

978-1-4244-7818-7/10/$26.00 ©2010 IEEE

Padmanabhan and Mogul [1] recommended predictive model to be used as a server hint. The simulation shows the reduction of latency until 45%. However, the network traffic even worse two times compare to the conventional solution [1]. Therefore, there is a need to provide justification of why it is important to cater for the issue on latency and network reduction simultaneously.

Bestavros *et al.* [9] presented a model for the speculative dissemination of World Wide Web (WWW) data. The work illustrates reference patterns from a Web server that can be used as the main source of information for pre-fetching. It shows that latency reduction increases until 50%, though it still increases the bandwidth utilisation. On the other hand, Pallis *et al.* [10] provide the solutions on pre-fetching by implementing a clustering method. The technique creates the cluster for pre-fetching to be graph-based clustering that consists of correlated and directed clusters. This technique can be adapted in a Web proxy server to increase the performance. The simulation conducted on a real data set implies that the technique is able to increase performance of the Web caching environment efficiently.

Kroeger *et al.* [3] suggested that a local proxy caching can decrease latency up to 26%, while pre-fetching could decrease latency up to 57%. The combination of both will give better latency reduction until 60%. Furthermore, the study also describes that algorithm on pre-fetching contributes to a reduction of latency, and it can provide double improvement on caching. However, it only happens when we decrease the latency.

Harding *et al.* [8] studied pre-fetching on a mobile device by enhancing the feature towards optimum utilization. This feature allows users to connect to the server when required and only a few amount of data need to be transferred. When a number of tasks and services have been done in desktop application, the mobile environment must be synchronized and customized when it requires more data. However, the method still faces challenges in the aspects of an application design, user interaction, usability and mentality model. Ye *et al.* [11] created a model that manages Web caching in a multi-device that is connected to two kinds of wireless network: MSS (Mobile Support Station) and the neighbouring peers to form the P2P network.

A number of researchers have started searching and observing the way to make an easier interaction between mobile devices and servers. Harding *et al.* [8] proposed a novel architecture for communication between a mobile device, and its server by adding a feature for preparation of data at the mobile device. It improves the communication that becomes almost similar to that of desktop applications. This feature acts as preparation tools for users as they only communicate with a server once the data is ready. The data that will be transferred over to a mobile device and a server will decrease the bandwidth load. Thus, it enhances the communication speed between them.

III. Intelligent Mobile Web Pre-fetching (IMWeP)

This paper proposes the Intelligent Mobile Web Pre-fetching (IMWeP) technique that calculates the number of hits for each Facebook's feature during users' browsing activity by automatic classification of user most favourites. IMWeP records each activity and restores the activity during the second login. Fig.1 demonstrates the architecture that implements the proposed technique.

The proposed technique calculates the cumulative number of hits for each Facebook Mobile feature (inbox, home, profile, and friends) when the user visits the feature. The proposed architecture consists of pre-fetching on mobile environment to increase the access of social network like Facebook using mobile devices. The pre-fetching based on XML is installed on a mobile device in order to handle the communication between a client and its server. This mechanism, also known as Social Network Concept, has the ability to handle and prepare data for client server intelligently. This smart client will control the schedule to access to the server. The feature on a social network is classified into a priority based on their popularity.

Fig.2 illustrates the flow of users' activity when using IMWeP system that applies the proposed technique. It starts by login into IMWeP. Then users will login to Facebook Mobile application for the first time due to the uncreated pre-fetching file. After the user browses and exits the application, the pre-fetching file will be saved into the storage of a mobile phone and will be loaded when users login into IMWeP in the future.

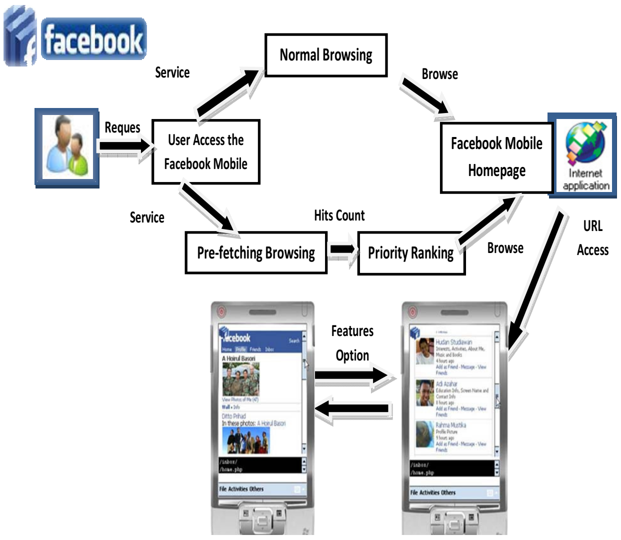


Figure 1. Architecture of Intelligent Mobile Web Pre-fetching (IMWeP) adopted in the Facebook Mobile application.

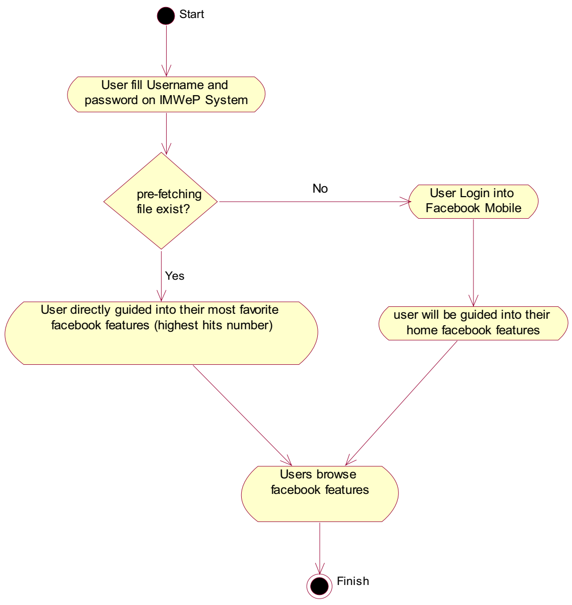

Start
User fill Username and password on IMWeP System
pre-fetching file exist?
No
User Login into Facebook Mobile
Yes
User directly guided into their most favorite facebook features (highest hits number)
user will be guided into their home facebook features
Users browse facebook features
Finish

Figure 2. Flowchart for users' activity in IMWeP proposed system.

3.1 Pre-fetching Using XML

Said *et al.* [6] proposed a pre-fetching concept based on XML database. The client will do "cold data pre-fetching" on a mobile device by creating a query generator. On the other hand, our work focuses more on providing, offsetting and saving the data on a mobile client to increase the performance in accessing the server. Each feature of Facebook Mobile that a user visits will be calculated and the hit number is saved into an XML file. This XML file will be read when users log on after the first login. The IMWeP system will read the XML file pre-fetching, and it will directly go to the most popular or favourite features visited by the user of Facebook Mobile. For example, if home/wall has the highest hit for a user, then the system will direct the user to the page automatically. The XML file is saved according to the username that is provided by the user when starting the IMWeP system.

3.2 Computation of the Favourite Features

In this study, we proposed technique computes the sum of hits as below:

$$hit\_inbox = \sum_{i=0}^{n} x_i ... x_n \tag{1}$$

This formula is also applied to compute the hit for other features: home, profile and friends. For each hit, it will be saved into XML file, and the cumulative consumption time will be recorded into XML file.

$$hit\_home = \sum_{i=0}^{n} h_i ... h_n \tag{2}$$

$$hit\_friends = \sum_{i=0}^{n} f_i ... f_n \tag{3}$$

$$hit\_profile = \sum_{i=0}^{n} p_i ... p_n \tag{4}$$

Where x_i, h_i, f_i and p_i are the hits for each Facebook Mobile feature that will be accumulated every time a user visits the feature. The classification aims to set up priorities among the features in a social network application or its elements in order to balance the network traffic.

Our research chooses Facebook Mobile as the case study as it is the most popular site among existing social networking Websites. In order to calculate the loading time for each feature, the URL is identified by using Regex function as shown in Fig.3. The algorithm receives a keyword for searching, and it continues with the matching process of the keyword versus the URL input. When the string keyword matches with one of the words in the URL input, the process continues with the counting of number of hit and duration, and saves the URL of the matched keyword.

```
Input: URL
Process:
    1.  Insert keyword for searching (eg:"home")
    2.  Search for the word that match with
        keyword.
    3.  If found match word
            a.  Counter up hit the keyword
            b.  Save the URL keyword
            c.  Counterup the duration
    4.  Else Nothing
    5.  End
```

Figure 3. Regex function.

The classification on a social is to set up priorities among social network features or elements to balance up the network traffic. The URL is identified into Facebook's features by using Regex function (Fig.3) to compute loading time for each feature. Fig.4 shows the hit number for each Facebook feature, for example "*you visit home:2x*". This statement means Regex function calculates a user visited "*home*" two times during the browsing. The hit number is grouped in tree-view illustration to give a clear and easy view to a user.

IV. EXPERIMENTAL DESIGN OF THE PROPOSED IMWEP

The experiment aims to collect the latest information about favourite features among the users of Facebook Mobile using IMWeP system. Thirty seven users among the undergraduate of the bachelor's degree who are in the fourth semester were recruited to test the functionality of IMWeP system. At the end of this study, we used a keystroke-level model to compare the performance of the loading process for

Facebook Mobile features either use IMWeP or without a use IMWeP.

4.1 *Assessment of IMWeP Features*

Before the users assess the system, they were asked for their favourite features when using Facebook Mobile. Table 1 shows that the Facebook feature that they like most is home/wall (68%), followed by profile (14%), friends (11%), and others (8%).

TABLE I. FACEBOOK FAVOURITE FEATURES

Facebook Features Mostly Liked					
Facebook features	Home/wall	Profile	Inbox	Friend	Others
User response	25	5	0	4	3
Percentage	67.5%	13.5%	0.0%	10.8%	8.1%

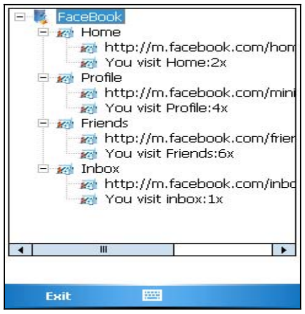
In addition, Facebook's features on home/profile consume more time in terms of visiting time compared to other features (51%), followed by other features such as games (30%). The rest belongs to friend features (11%), and profile (8%), see Table 2.

TABLE II. FACEBOOK FEATURES TIME CONSUMPTION

Facebook Features Spend Most Time					
Facebook features	Home/wall	Profile	Inbox	Friend	Others
User response	19	3	0	4	11
Percentage	51.4%	8.1%	0.0%	10.8%	29.7%

4.2 *Browsing Tasks Using IMWeP*

Each user will be requested to run IMWeP system and then browse Facebook Mobile. IMWeP will record and count the hit for each Facebook Mobile feature. The user can view the history of hit based on the features in the tree-view illustration provided by IMWeP. Fig.4 shows the tree-view that provides the user with their favourite information. The highest hit number of the features will be ranked on the first priority of loading process when the user logs in.



```
FaceBook
  Home
    http://m.facebook.com/hon
    You visit Home:2x
  Profile
    http://m.facebook.com/mini
    You visit Profile:4x
  Friends
    http://m.facebook.com/frier
    You visit Friends:6x
  Inbox
    http://m.facebook.com/inbc
    You visit inbox:1x
```

Figure.4 Tree-view feature in IMWeP system.

Besides the tree-view, IMWeP also records the number of hit for each feature in XML file. The results show that most users have interests in viewing their profiles and followed by clicking home. Home features have the second highest hit due to users often read and write on the wall. See the comparison of hits for each feature in Fig.5.

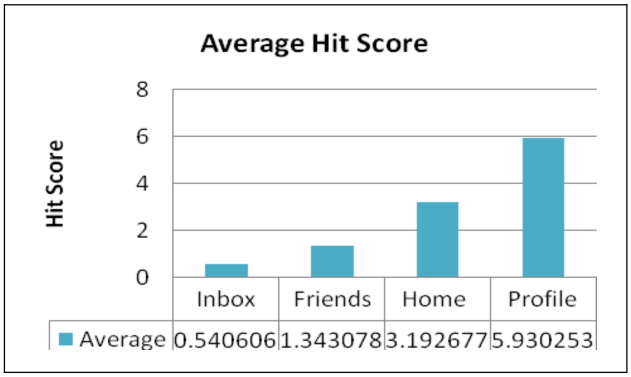


Figure 5. Average hit score.

The pre-fetching of XML file, records the users' activities during their interaction on Facebook. The data are saved in XML file as a number of hits, cumulative time, and the total cumulative time. Time consumption is computed from the loading time of the browser upon clicking one of Facebook's features (see Table 3 for the details).

TABLE III. AVERAGE OF TIME CONSUMPTION USING IMWeP

Average Time Consumption	
Average cumulative time Inbox	0.27
Average cumulative time Friends	1.87
Average cumulative time Home	6.62
Average cumulative time Profile	8.97
Average cumulative time total	**17.73**

Table 3 portrays that profile has the highest cumulative time while home becomes the second longest cumulative time. The lowest is an inbox cumulative time. This mean profile has made user to spend a lot of time due to a user tends to view and edit their profile frequently.

The cumulative home and profile time are greater than others due to users' activities as they access home and profile features more often. At the end of the test, the users were asked to fill in the questionnaire regarding their experiences on using IMWeP system while browsing the features. 41% agreed on the usability of the tree-view menu while 19% strongly agreed that tree-view was very useful to help users in navigating the Facebook features. Refer to Table 4.

Fig.6 shows the evaluation result of IMWeP based on users' experience while interacting with IMWeP to browse through Facebook's features. 18 users from 37 respondents agreed that IMWeP is a very useful system, and it is comparable with the regular Facebook application.

TABLE IV. USABILITY ON TREE-VIEW

Usability of Tree-view	
Rating	**User Response**
1(strongly disagree)	1(2.7%)
2(disagree)	2(5.4%)
3(Moderate)	12(32.43%)
4(Agree)	15(40.5%)
5(Strongly Agree)	7(18.9%)
Total	**37**

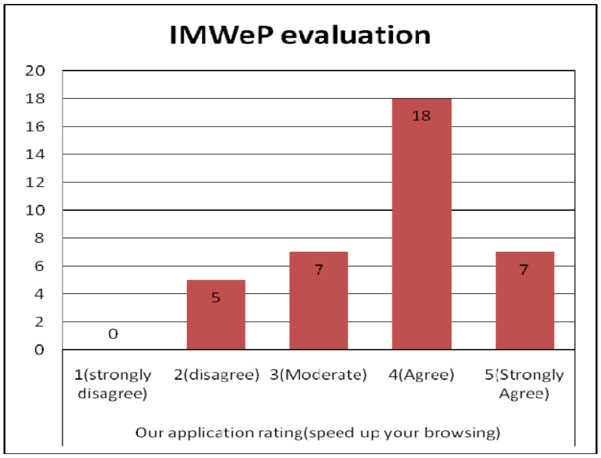


Figure 6. Users evaluation on the IMWeP system.

Another experiment was conducted to compare the execution time of Facebook Mobile using IMWeP and without IMWeP system. A keystroke-level model was used to compare the performance. Keystroke-level model is proposed by Card *et al.* [12]. It also has been discussed by the other researchers [13][14].

This paper introduces three kinds of scenarios that have been applied during the experiment:

A. First Scenario→ User logs in Facebook Mobile without IMWeP system, and the home feature will appear. Next, the user will continue to browse through other features. For example, user A wants to see its profile. The action sequence of this scenario that is executed based on Kieras [14] can be seen and described as below:

Fact:

Average non-secretariat typist (40 wpm): 0.28 sec [14]
User A, username:A@mail.com
Password:A

In IMWeP system and Facebook Mobile, users do not have to insert their username and password. It is assumed that the users use the same username and password. This means it saves the time to write username and password and also click the login button. As explained in a third scenario on point C, that user does not need to insert Facebook Mobile password again. This is the reason why the computation time for username and password execution time is multiplied by two in first scenario because they have to type their username and password each time they want to use Facebook Mobile.

Action Sequence

```
1. Point to username textbox
2. Write  username
3. Point to password textbox
4. Write password
5. Point to login button
6. Click the profile link feature
```

Operator Sequence

```
1. Point to username textbox P
2. Write  username(Assume 4 word)
3. Point to password textbox P
4.  Write password (Assume one word)
5.  Point to login button P
6.  Click login button B
7.  Point to profile link P
8.  Click the profile link feature B
    Total Time =2*(4P+(4*0.28)+0.28)+2B
                 =2(4*1.1+1.4)+2*0.1
                 =11.8 Second
```

B. Second Scenario→ Users open the tree-view menu and check the information and click on the link they want to visit.

Action Sequence

```
1. Point to tree view menu
2. Click tree view menu
3. Point to profile feature
4. Click the profile feature
```

Operator Sequence

```
1. Point to tree view menu P
2. Click tree view menu B
3. Point to profile feature P
4. Click the profile feature B
    Total Time =2P+2B
               =2*1.1+2*0.1
               =2.4 Second
```

C. Third Scenario→ For the first time of login, a user must complete several steps as shown in the first scenario. However, for the second and the following logins, the user only requires to fill in the username and password in the IMWeP system, and they will be directed to the profile pages (assume the profile has the highest hit number).

Action Sequence

```
1. Point to username textbox
2. Write  username
3. Point to password textbox
4. Write password
5. Point to login button
6. Click login button
```

Operator Sequence

```
1. Point to username textbox P
2. Write  username(Assume 4 word)
3. Point to password textbox P
4. Write password (Assume one word)
5. Point to login button P
7. Click login button B
   Total Time =3P+ (4*0.28) +1*(0.28) +B
                 =3*1.1+1.4+0.1
                 =4.8 Second
```

The result shows that the proposed IMWeP system in Facebook Mobile proves to be faster than Facebook Mobile without such pre-fetching tool and tree-view features. See Fig.7 for the illustration in graph mode the access time comparison between three different scenarios.

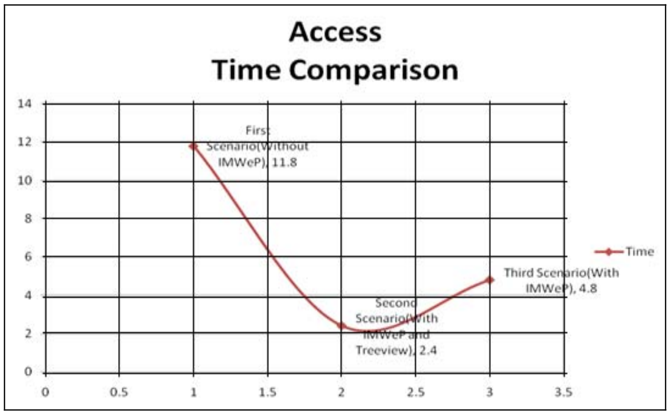


Figure 7. Time comparison between using IMWeP and without IMWeP.

V. Discussion and Conclusion

This paper has discussed Intelligent Mobile Web Pre-fetching (IMWeP) technique that can increase the speed in accessing information via a mobile environment. Based on the experiment, Web pre-fetching is anticipated to contribute in the performance of communication with the server from the client side. The experiment shows some scenarios to prove the performance of IMWeP technique as compared to common Facebook Mobile. It shows the improvement up to 40.7% (4.8 second in comparison to 11.8 second). IMWeP technique is faster than normal browsing technique without pre-fetching. This is also supported by more than half (68%) of the respondents agreed that IMWeP system was very useful and had improved the speed in accessing and opening Facebook Mobile. The finding encourages us to combine Web caching on a proxy in the future work.

Acknowledgment

The authors would like to thank the Ministry of Science, Technology and Innovation Malaysia (MOSTI) for the e-science fund vote 79330. The first author thanks the Ministry of Higher Education (MOHE) for the scholarship award. An appreciation also goes to Research Management Centre (RMC) and Human Capital Development Unit, Universiti Teknologi Malaysia (UTM) for their overwhelming cooperation. We are sincerely thankful to Ahmad Hoirul Basori for providing helpful and excellent support to realize this research.

References

[1] Padmanabhan, V.N. and Mogul, J.C., "Using predictive pre-fetching to improve world wide web latency", Computer Communications Review,vol. 26, pp. 22–36, July 1996 ACM, SIGCOMM'96.

[2] Davison, B.D.,"Adaptive Web Pre-fetching", Proceedings of the 2nd Workshop on Adaptive Systems and User Modeling on the WWW, 2001.

[3] Kroeger,T. M., Long, D. E. and Mogul, J. C., "Exploring the Bounds of Web Latency Reduction from Caching and Prefetching", USENIX Symposium on Internet Technologies and Systems (USITS), Monterey, CA, December 8-11, 1997.

[4] Maltzahn, C., Richardson, K. J., Grunwald, D. and Martin, J. H., "On Bandwidth Smoothing", Proceedings of the Fourth International Web Caching Workshop (WCW99), San Diego, CA, March 31-April 2, 1999.

[5] Sulaiman, S., Shamsuddin S.M.,and Abraham, A. "Rough Neuro-PSO Web Caching and XML Pre-fetching for Accessing Facebook from Mobile Environment", 8th International Conference on Computer Information Systems and Industrial Management (CISIM 2009), IEEE Press, ISBN: 978-1-4244-5612-3, 2009, pp. 884-889.

[6] Said, E.G, Omar, E.B., and Robert,R., "Data Pre-fetching Algorithm in Mobile Environments", European Journal of Scientific Research, ISSN 1450-216X Vol.28 No.3 (2009), pp.478-491.

[7] Hong M., Ryu, D-Y., Sir, J-C., Kim, E-Y. and Lim,Y-H., "Using a Transcode and Prefetch Method for Playing XML Contents Containing Multiple Multimedia Data on Mobile Terminals", EDBT 2004Workshops, LNCS 3268, pp. 309–317, 2004.

[8] Harding, M., Storz O., Davies, N. and Frida, A., "Planning Ahead: Techniques for Simplifying Mobile Service Use", HotMobile 2009, Santa Cruz, CA, USA.ACM, February 23–24, 2009.

[9] Bestavros, A. and Cunha, C., "A pre-fetching protocol using client speculation for the WWW", Tech. Rep. TR-95-011, Boston University, Department of Computer Science, Boston, MA 02215, Apr. 1995.

[10] Pallis, G., Vakali, A. and Pokorny, J., "A Clustering-based Approach for Short-term Prefetching on a Web Cache Environment", Computers & Electrical Engineering Journal, Elsevier, 34(4), 2008, pp. 309-323.

[11] Ye, F., Li, Q, Chen, E., "Adaptive Caching with Heterogeneous Devices in Mobile Peer to Peer Network", SAC'08, March 16-20, 2008, Fortaleza, Ceará, Brazil.AC M.

[12] Card, S. K., Moran, T. P., and Newell, A.,"The psychology of human-computer interaction", Hillsdale, NJ: Lawrence Erlbaum Associates, 1983.

[13] Olson, J. R., and Olson, G. M., "The growth of cognitive modeling in human-computer interaction since GOMS", Human-Computer Interaction, 5, 1990, pp. 221-265.

[14] Kieras, D, "Using the Keystroke-Level Model to Estimate Execution Times", University of Michigan, 2001.