Abstract

In spite of the great popularity which mobile devices have in our days, the Internet access through these devices is extremely limited. This must to that Web sites have not been developed taking into account the characteristics and limitations of mobile devices. Thus for example, screen restrictions, low capacity of memory and storage, high costs of Internet connection and the deficient bandwidth; among others, they have restrained the visualization of Web sites through mobile devices. This work is tried to increase the visualization of Web sites on mobile devices improving the user’s navigation experience, on the one hand, handling different formats from the resources and by another one, handling to a communications scheme more efficient for exchanging information, which with takes to diminish costs of accessing to the Web by air-time connection or by volume of information, given that the size of resources is diminished and the access time is made agile.

1. Introduction

In the past (50s, 60s and half-full of 70s), the computer science paradigm was: “one computer, many users”.

At the end of the 70s and principles of the 80s, the computer science paradigm change to “one computer, one user”; due to the appearance of the Personal Computers (PC).

At the end of the 80s and in the 90s, the computer science paradigm change a little due to the appearance of computers networks and mainly to the appearance of the Web and popularization of the Internet.

Because we are a society in movement where the only constant is the change and that the Web has become a massive mean of communication, it is needed to access to the information available in the Web at any time, any place and through any mean. One of the technological advances that have made possible all this is, without a doubt, mobile devices. This bringing as consequence a radical change in the computer science paradigm, reason why at the beginning of 2000 year, we start to speak about the new computer science paradigm: “one person, many computers”.

The great majority of indicators like [1], [2] and [3]; among others, indicates the great potential that have mobile devices. Nevertheless, the use of Internet on mobile devices is extremely limited [4]. This must to the high costs to access the Web and others problems that mobile devices present, which have taken to a bad navigation experience for users and they choose alternating methods for access to the Web.

The main problems that present mobile devices are the following ones (see Figure 1):

1. Deficient methods of information entrance (small keyboards if they exist, ineffective handwriting recognition, etc.).

2. They count on few resources in comparison with a PC (low memory, low storage space, few peripherals, slower microprocessors, etc.).

3. Finite energy provision (between more capacities of processing, smaller capacity of battery duration).

4. Because they use wireless networks and they have high mobility, the disconnections are frequent. In addition, they have low bandwidth and in general terms, they have poor Quality of Service (QoS).

5. The information displaying is limited because they have smaller screens with low resolution and colors.

The last two problems are those that they have more influenced in the low use of mobile devices to access to the Web, and are those that in this project we focused to solve.
The Web needs a persistent connection. That is to say, if a connection does not exist or disconnections appear frequently as in the case of mobile devices, the access to Web resources cannot be realized. It has been detected that the client/server model on mobile devices does not follow the traditional interactive synchronous model, mobile devices follows an asynchronous non-interactive scheme in where the users are not connected all the time.

The problem that appears when having smaller screens is than it is needed more user interaction. Between this problem, it emphasizes the scrolling problem, where the user must cross towards in all screen directions with the objective to visualize the Web resource in complete form (see Figure 2). Another problem consists that many Web browsers on mobile devices do not support some characteristics of the pages, reason why they are not possible to be visualized or they visualize in incorrect way.

In order to solve the Web content visualization problem on mobile devices, it is needed on the one hand, a mechanism that allows to visualize Web resources without concerning the connection state, this mechanism receives the name of hoarding. On the other hand, it is needed a mechanism that allows to adapt the Web content on the Web to the constraints and characteristics of mobile devices, this mechanism receives the name of transcoding.

2. Concepts

2.1. Mobile devices

Mobile devices are electronic devices that are used for the communication, processing and interchange of data and can be taken by their users to send, receive or share data with other devices. The characteristics of this equipment are the following ones according to [5]:
1. They are possible to be transported easily.
2. They count with wireless network interfaces.
3. They have electrical autonomy.
4. They have RAM memory for storage and programs execution.
5. They have forms of permanent storage.

At the moment, exist a great variety of mobile devices, which can be classified depending on its hardware and software characteristics, so and as they are at Figure 3.

The operating systems on mobile devices are divided in three great groups: the based ones on Symbian, the based on PalmOS and the based on Windows Mobile [6]. Recently devices based on the Linux operating system have been used for mobile and embedded systems [7].

2.2 Hoarding

Hoarding can be defined as the replication process and processing in disconnection mode of data previously selected and copied locally in the mobile client [8]. The hoarding scheme consists of the following steps:
1. Identification of access patterns.
2. Selection of the resources that will be replicated.
3. Reintegration Control of replicas.

Hoarding is based on the idea that if the user cannot access to the data, the data follow the users. That is to say, during periods of low activity, Web sites are sent to the mobile devices that these maintain in its cache and thus can visualize Web sites in asynchronous mode or in disconnection mode. With this it is avoided that the user must connect itself whenever needs a resource Web.

The problem of this mechanism consists in determining that information of the Web site is necessary to replicated to mobile devices, since it has been mentioned, mobile devices have a reduced enough storage space.

This process is solved through applying Web use mining to Web servers or Proxy logs to identify access patterns. An access pattern indicates the possible resources that a mobile user could need in a Web site with a greater or equal probability to 80%. The identification of the access patterns is made by means association rule algorithms [9]. Once identified the patterns it is come to trim the Web site as one is at Figure 4.

![Figure 4 General scheme of the hoarding process (it trims of a tree pattern).](image)

Once trimmed the tree, it is come to download the resources, compress and send them to mobile devices.

2.3 Transcoding

The mechanism of Web content transcoding, carries out a reorganization and grouping of the elements contained in the Web page according to the boundary of the entrance language (HTML). Like final result of the transcoding are Web pages are whose presentation or visualization format is optimal for a device with limited displaying, trying to faithfully respect the original semantics (structure) of the page [10].

3. Antecedents

Until the moment have not been works that hoarding and transcoding Web in joint way. It exists works that on the one hand make Web content transformation and works that make replication of Web sites. In [11] we reported works related on the transcoding area as well as we described more extended the transcoding process of Web sites.

In [12], there is a tool that allows transforming a HTML resource to adapt to mobile clients. The transformation is made in a single format HTML and is not transparent for the users, since they must put to a portal and introduce the URL of the resource to transform.

In [13], there is a work in which Web sites are replicated based on levels (reason why they are possible to be had much resource). Transformation is not made online since the Web sites are designed specially for this tool and can be visualized by many devices that have the client version of this program.

In [14], there is a work in where Web sites are cut in which the original page is conserved and single those parts that are modified constantly are updated. It is necessary to trim these pages manually so that they are available for others clients.

In [15], there is a tool (like many others in the market exist) that transforms documents from a format to another one (Word to PDF, etc.). The transformation is made in a desktop PC and later the modifications are synchronized with the mobile device client program.

In [16] and [17], there are related works of programs that make Web sites replication and allow to work in disconnection mode. The difference is that they do not discriminate the resources to replicate in optimal way.

4. Methodology of solution

The solution scheme that we developed consists of an adaptation of the client/server scheme oriented to mobile clients. This model consists as much of mobile clients as of a servers in charge to offer services of Web resources. In the middle of our clients and servers is our layer of intermediaries, at the client side and at the server side.

The Proxy at the client side receives the name of GAP (Hoarding Manager for Pocket PCs devices) and the server side proxy receives the name of GAT.
(Hoarding and Transcoding Manager). The system architecture is showed at Figure 5.

![Figure 5 General solution model.](image)

The GAT is divided in two main components. MA (Hoarding Mechanism) in charge to manage the patterns from access to Web sites to be able to replicated them on mobile devices and MT (Transcoding Mechanism) in charge to transform a page from one format to another one in view of its correct visualization into mobile devices.

GAP on the other hand, is in charge to manage the disconnection events that could be presented on mobile devices, reason why hoards resources in the local device cache. GAP is conformed basically of three main modules: Observer, Local Hoarding Manager (GAL) and Local Disconnection Manager (GDL).

The Observer is in charge to process each request and to give back the result to the browser. The GAL is in charge of the manipulation and control of the cache in the device. It is the user who decides what resources are desired to hoarding, as well as to limit the storage space. The GDL is in charge to determine the state of the connection.

GAT MT is basically in charge to interpret the format that contains the Web resource request, reviewing the X-Transform label (that GAP generate) and if it is a mobile device like Windows CE (Pocket PC, Smartphone), PalmOS (Palm, Treo, Clie), EPOC and SymbianOS, transcoding the page.

The general operation of the system is the following one. The user introduces a URL from its browser (who previously has been configured to redirect his exit towards GAP). GAP receives the request and determines if the resource is in the local cache of the device, if it finds the resource, sends the hoarded resource to the Web browser.

When the resource is not hoarded, GAP validates that connection exists and it is to obtain the resource on line. If for some reason the resource cannot be shown, (or that does not exist or it is had detected an error in the connection) notifies to the user sending an error message.

On the other hand, if the Web resource is not hoarded and a pattern of the site in the local device does not exist, the MA sends the Web resources if the pattern access for this site exists. If the pattern exists but the resources hoarded in the MA are not in the cache, this one obtains them requesting to MT and soon it compresses the resource in ZIP format to optimize the downloading process.

Once the MA has sent the hoarded Web site, the mobile device must decompress the Web site and update its list of patterns. This process happens in transparent way without the user perceives it.

MT is in charge to collect documents and in the case of being HTML resources; it transforms them if it is that the configuration parameters therefore indicate it. The transcoding is made online, reason why the process is slowed down if the document is too weight.

The Figure 6 shows the GAP execution on Windows Mobile devices. On the superior part, on the left side is the message when connection does not exist and the resource is no hoarded; whereas in the right side, they are a Web page with hoarded resources (note that the images in the original resource are different from the shown ones, since they were replaced in the cache to exemplify this process). On the inferior part, there are images of the prototype working in which it can among other things to visualize the state of requests as well as to configure the system.

Although at the moment, GAP is only executed on mobile devices, the transformation can be made in a great diversity of heterogeneous mobile clients as well as traditional clients. The only indispensable requirement is that they count with a Web browser which allows redirecting its exits towards a Proxy server.

Figure 7 shows the form in which the transformation of the Web resource is made. The process consists on extracting elements with similar characteristics of the site and to group them to give a reformat. In our work all the text elements are grouped, folled of the images and the links. On the other hand, Figure 8 shows a example that how a transformed resource can be visualized in diverse formats. With these procedures we guaranteed that the Web site can be acceded by a great variety of mobile clients.

5. Results

Several tests with the developed prototype were made to evaluate their performance. The variables that
interest to us to know are the performance of the battery, times of processing, as well as the size of the resources. On the basis of these variables the following hypotheses were formulated:

\( H_1 = \) with the use of the prototype and in general of the hoarding process, the energy consumption of the device diminishes considerably because we work on asynchronous and disconnection mode, reason why the consumed energy is minor when having turned on the wireless network interface.

\( H_2 = \) with the transcoding process obtains in general Web resources minor since when reformatting some elements of the page its size is reduced.

\( H_3 = \) with the hoarding process is considerably reduced the size of the Web sites when making a cut of the Web site on the basis of a hoarding pattern.

\( H_4 = \) the access time to Web resources is significantly reduced when having resources hoarded on mobile devices.

The tests consisted taking into account a set from 100 Web sites taken from an electronic mail survey to many users. For some cases the test was reduced up to 29 sites because many resources could not be transformed well.
approximated time of 52 minutes. The characteristics that had the mobile devices to normalize the tests consisted of not having users process executing (only system processes Web browser and GAP), the screen brightness to maximum intensity, the saving energy and hibernation options disabled, and the wireless network interface enabled. Once obtained the consumption of energy without visualizing Web sites, the system was come to register the consumption of energy being worked but without doing use of hoarding and later with hoarding, obtaining the following increases in the energy consumption (see Table 1).

Table 1 Obtained results visualizing sites hoarded and without hoarding.

<table>
<thead>
<tr>
<th>Platform</th>
<th>Without hoarding</th>
<th>With hoarding</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>PC</td>
<td>+9</td>
<td>+1.5</td>
<td>+7.5</td>
</tr>
<tr>
<td>SP</td>
<td>+7</td>
<td>+7</td>
<td>0</td>
</tr>
<tr>
<td>PPC</td>
<td>+14</td>
<td>+4</td>
<td>+10</td>
</tr>
</tbody>
</table>

We obtain an energy consumption average of 8.75% minor using hoarding that without using it. This is the reason why \( H_1 \) is certain.

With respect to \( H_2 \), it was come to transcoding Web resources in different formats to see the size of the transcoding resources (in [11] we were centered in seeing the times of the transcoding process). The obtained results are showed in Table 2.

Table 2 Size of the transcoding resources.

<table>
<thead>
<tr>
<th>Format</th>
<th>Size(KB)</th>
<th>% Saving</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML</td>
<td>17.31</td>
<td>0</td>
</tr>
<tr>
<td>HTMLR</td>
<td>8.89</td>
<td>48.58</td>
</tr>
<tr>
<td>WML</td>
<td>6.94</td>
<td>59.87</td>
</tr>
<tr>
<td>XHTML-MP</td>
<td>7.19</td>
<td>58.43</td>
</tr>
<tr>
<td>PDF</td>
<td>14.75</td>
<td>32.19</td>
</tr>
<tr>
<td>PS</td>
<td>17.31*</td>
<td>0</td>
</tr>
<tr>
<td>XML</td>
<td>17.31*</td>
<td>0</td>
</tr>
<tr>
<td>TXT</td>
<td>7.67</td>
<td>55.66</td>
</tr>
</tbody>
</table>

It is observed that a considerable reduction in the size of the transformed resources exists, does not happen the same in some cases in special in format like PS and XML in where due to transformation errors the size of the resources went off enormously influencing the results, reason why we decide to omit them. An average of 33.9% size reduction of transcoding resource was obtained. Due to this reason \( H_2 \) is certain.

For \( H_3 \), the hoarding was obtained based on access patterns of some Web server logs and in other cases with synthetic patterns. We obtained the results showed at Table 3.

Table 3 Results obtained when hoarding and transcoding Web sites in different formats.

<table>
<thead>
<tr>
<th>Format</th>
<th>Size(KB)</th>
<th>% Compression</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTML</td>
<td>2292.50</td>
<td>0</td>
</tr>
<tr>
<td>HTMLR</td>
<td>1261.83</td>
<td>44.95</td>
</tr>
<tr>
<td>WML</td>
<td>1723.43</td>
<td>24.82</td>
</tr>
<tr>
<td>XHTML-MP</td>
<td>1554.52</td>
<td>32.19</td>
</tr>
<tr>
<td>PDF</td>
<td>1539.46</td>
<td>32.84</td>
</tr>
<tr>
<td>PS</td>
<td>1886.51</td>
<td>17.70</td>
</tr>
<tr>
<td>XML</td>
<td>1323.35</td>
<td>42.27</td>
</tr>
<tr>
<td>TXT</td>
<td>1164.83</td>
<td>49.18</td>
</tr>
</tbody>
</table>

Thanks to hoarding Web site in average we can reduce 34.85% the size of a Web site, and if we combined this with a ZIP compression algorithm, we are possible to be obtained improvements until of 86.62%. On the basis of these tests one concludes that \( H_3 \) is certain.

For \( H_4 \) the times were taken through the GAP logs when having hoarded Web sites on the device. The average to access to a Web resource online is of 2,509.76 milliseconds, whereas to obtain resources in the cache of the device required 368.82 ms, obtaining an access of 85.30% more rapid. Due to this reason we demonstrate that \( H_4 \) is certain.

It was observed that tests on the basis of the size of the hoarded Web sites had been made but not on the basis of the time in which they take in hoarding itself, reason why it was again come to modify the system to register the times of hoarding. The results are showed at Table 4.

Table 4 Hoarding times in HTML and HTMLR (HTML Reformatted).

<table>
<thead>
<tr>
<th>TAHTML</th>
<th>TAHTMLR</th>
<th>%Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>63290.9567</td>
<td>22460.0385</td>
<td>281.793626</td>
</tr>
</tbody>
</table>

The times are in milliseconds reason why the average time of hoarding without transcoding (TAHTML) is 22.46 seconds; whereas the hoarding time of a Web site that has been transcoding in HTMLR format (THMLR) is 63.29 seconds which does 281.79% slower.

6. Conclusions

With the present work we have a software platform that is able to allow visualizing Web sites without concerning the characteristics of mobile devices.

We obtained a transparent adaptation of Web applications to the client/server architecture on mobile
environments. With which it is avoided to have to modify existing applications and protocols.

In addition, with the prototype presented here, it is able to visualize Web resources in transparent form for the users, since it is possible to transform Web documents (HTML) on-the-flight to other formats like WML, XHTML-MP, PDF, XML, Postscript, plain text and HTML reformatted for mobile devices.

Thanks to hoarding and transcoding mechanisms, we guarantee the correct Web sites visualization on mobile devices, it is possible to save costs since the resources size is reduced with hoarding in average 35% of a Web site size and if compression of a Web site is made can arrive as large as reach a reduction to of 86% Web site, the transcoding process in average reduces to 34% the size of a Web page, we have a light energy saving of the batteries on the devices, as well as thanks to the local cache on the mobile device that can be made more agile the access times to the resources until a 85%.

In order to understand better these obtained results, an example of the visualization of 70KB Web page visualized through Mexican cellular telephony provider Telcel which prices in networks based on air time of connection CSD is $1.5 Mexican pesos per minute; and by volumes of information GPRS (transferred 0.12 KB) is showed at Table 5. The prices doesn’t include taxes.

### Table 5 Saving costs of connection to the Web using schemes of hoarding and transcoding.

<table>
<thead>
<tr>
<th>Tarea</th>
<th>Tamaño (KB)</th>
<th>Time (Sec.)</th>
<th>GPRS</th>
<th>CSD</th>
</tr>
</thead>
<tbody>
<tr>
<td>70 KB Web page</td>
<td>76.1</td>
<td>455</td>
<td>$9.24</td>
<td>$12</td>
</tr>
<tr>
<td></td>
<td>TRANSCODING</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>70 KB Web page</td>
<td>50.22</td>
<td>300.26</td>
<td>$6.12</td>
<td>$9</td>
</tr>
<tr>
<td></td>
<td>HOARDING</td>
<td></td>
<td></td>
<td>$3</td>
</tr>
</tbody>
</table>

7. Further works

As future works we propose the implementation of the GAP module in other platforms doing use of J2ME technology. This with the purpose of includes another platform of mobile devices.

Another work that is tried to make consists of handling a totally asynchronous mechanism to receive Web sites using SMS/MMS technology. In this work, the user will introduce in a SMS message the URL, the transformation format, among other characteristics; later the user receives in a SMS message the response to its transcoding page.

Another additional work consists of extending the transformation policies to allow that users can choose a better way to transform the Web resources.

Finally, we think to design a new mechanism for the identification of access patterns that are more efficient in real time to eliminate the limitations that at the moment this project has.

8. References