Smart Web Page Adaptation Systems For Mobile Web Browsing

P.John Augustine, Dr.K.Bommanna Raja and M.Veeramanikandan

Dept of Information Technology, Sri Eshwar College of Engineering, Coimbatore
Dept of Biomedical Engineering, PSNA College of Engineering and Technology, Dindigul
Dept of Computer Science & Engineering, Kalavai College of Technology, Coimbatore

Article history:
Received 4 September 2014
Received in revised form 24 November 2014
Accepted 8 December 2014
Available online 10 January 2015

Keywords:
Content adaptation, mobile device, rule base, page formatting, profiler system, mobile devices, existing content, adaptation systems, client device

Abstract
Background: Content adaptation has been playing an important role in mobile devices, wherein the content display differs from desktop computers in many aspects, such as display screens, processing power, network connection bandwidth. In order to display web contents appropriately on mobile devices, Mobile web browsing usually becomes time-consuming since currently it requires horizontal and vertical scrolling. Objective: users interested in only a section of a web page are often burdened with cumbersome whole web pages that not only do not properly fit their mobile screens but also require a lot of delivery time. This problem can be resolved with the help of a mobile web content adaptation system. Existing web content adaptation systems focus on resizing the contents to fit a mobile device and removing unnecessary contents from the adapted web page. This paper’s aim is to address the gap by proposing the new method that provides a condensed view of an adapted web page. In order to achieve this, the proposed algorithm will first categorize an HTML object. A depth-first traversal algorithm used to select the sequence of blocks to be displayed on a mobile device. Evaluate the usability of the adapted contents against other deployed systems. This enables users to locate targeted information within a web page in a shorter span of time.

Results: The proposed method is an automated mobile web content adaptation system. This increase the usability and friendliness of the webpage for the handheld devices.

Conclusion: this paper proposes new algorithms for adapting the layout of web content for mobile devices. To fulfill the objectives of the new solutions have been proposed to improve the efficiency and effectiveness for mobile web browsing.

© 2015 AENSI Publisher All rights reserved.

INTRODUCTION

Over the last few years, people started using network ready mobile devices like handheld computers, PDAs and smart phones to access the internet. The term mobile device refers to a device specially designed for synchronous and asynchronous communication while the user is on the move [Ramawamy, L., Iyengar, A., Liu, L., & Douglish, F. (2012)]. Among all the mobile computing devices, mobile phone is the most popular and mobile internet browsing is part of the ubiquitous computing trend. The term mobile device as a device specially designed for synchronous and asynchronous communication while the user is on the move. The term “Mobile Web” introduced when internet services for mobile devices started to use HTML based (W3C) standards for delivery. The mobile web refers to any mobile device like a smart phone or tablet computer connected to a wireless network by using an internet connected application or web browser. Among the mobile devices, mobile phones and PDAs are the most popular [Pathak, Y. Hu, and M. Zhang (2012)]. Moreover, one of the facilities, which contribute, to this popularity is the accessibility of the internet and more than six billion peoples all over the world are using web in the mobiles. However, mobile phones provide good mobility, but very limited computational capabilities and screen size.

Mobile web content adaptation refers to techniques that dynamically adjust the contents according to the properties of the handheld devices for better presentation [Hua, Z., Xie, X., Liu, H., H. Lu, & Ma, W. Y. (2013)]. Many different screen sizes are currently available on mobile phones. Typical mobile phone screen resolutions are 128x96, 160x120, 176x144, 208x176, 220x176, 320x240 and 352x288 pixels. The smart phone screen resolutions are 400x240, 480x320, 640x360 and 800x400 pixels. Accessing the same content from a
small-screen device (smart phones with screen sizes about 20 times lesser than that of a desktop screen) requires horizontal and vertical scrolling which is not comfortable for the user [Z. Wang, F. Lin, L. Zhong, and M. Chishtrie(2011)]. Therefore, small screens provide less visible opportunities at any given time, requiring users to rely on their short-term memory to build an understanding of web information.

The main purpose of this paper is to develop a mechanism to adapt the web contents for mobile devices. This paper focuses on the techniques needed to adapt all kinds of text contents and focuses on the techniques to change the html elements of the image, audio, flash and video contents on the web page so that the contents appear properly on the mobile screen [Yung-Wei Kao, Tzu-Han Kao, Chi-Yang Tsai., (2013)].

In this paper, our proposed technique adapts web contents to improve mobile web browsing. The new method first transformed into an html tree containing all the contents of a web page. The system then adds other functionality to the HTML tree and selects contents from the HTML tree. The most interesting technique of this framework is that it provides a condensed view of an adapted web page by toggle functionality. The toggle functionality added to each block title in a web page. This provides a condensed view consisting of only block titles that users can then expand into complete content.

Related Works:

Many types of mobile web content adaptation systems have been deployed but only a few of them have been commercialized for public use; these systems are not open source and the techniques the developers use to deploy the systems are not open to public [George Buchanan, Gary Marsden, (2011)]. On the other hand, in order to develop and improve the much needed mobile web content adaptation frameworks, researchers over the years have applied different types of experimental methods of which some are automated and some semi-automated. The next section discusses both the deployed and experimental systems in detail [van den Brande and A. Pras(2012)].

A. Deployed Web Content Adaptation Systems:

As mentioned earlier mobile web browsers, browsing requires a web content adaptation system for a better web browsing on the mobile phone. Many deployed adaptation systems not only contain the required adaptation features, but also serve all kinds of web pages for mobile devices [T. Maekawa, T. Hara and S. Nishio (2011)]. Skweezer is a deployed web content adaptation system. It adapts web pages designed for the desktop environment browsing and allows its users to access their favourite sites from mobile phones. It reduces bandwidth consumption because it removes the background colour and images. It automatically searches the web using Ask.com indexing. However, Skweezer is unable to adapt image files like png, tiff and gif, along with audio and video files. Therefore, these types of multimedia files don’t appear on the web pages after adaptation [Y. Chen, W.Y. Ma and H.J. Zhang (2003)]. Mowser is a deployed web content adaptation system that converts web pages in a predictable and linear manner so that the users can view the adapted web pages properly on mobile screens. In order to deliver productive web pages on a mobile device, it alters images and text format in certain aspects of web page functionality. Mowser adapts all kinds of web pages and delivers the contents in one-dimensional (1D) web pages. Therefore, the users need to scroll horizontally and vertically. After adaptation, some text information and links appear more than once while other texts and images don’t appear on a separate or new line [Dubey, P. De, K. Dey, S. Mittal, V. Agarwal, (2013)].

B. Experimental Adaptation Frameworks for Mobile Devices:

Many experimental web content adaptation systems have developed over the last few years like the CMO, Annotation-based Web document Framework, Xadaptor and Web Page Tailoring Framework [H.N. Chua, et al (2010)].

The CMO reduces information overload and thereby allows its users to see the most relevant fragments of the web pages and to navigate between fragments if necessary. The Interface Manager receives the request from the user, produces an html contents and delivers the requested page to the user [M. Butkiewicz, H. V. Madhyastha, and V. Sekar (2011)]. The Context Analyzer starts processing after the user uses the navigation. The Analyzer collects the context of the link, which relates to the links and text around the links. When a user navigates to a new page, the Context Analyzer uses the support vector machine to select the most relevant context from the new page and delivers it to the user. The Browser Object retrieves the contents of the web page and transfers them into a dome-tree. The Frame Tree Processor combines all the contents from the Dom-tree and renders it to the users [Y. Chen, X. Xie, W.Y. Ma and H.J. Zhang (2005)].

Annotation-based Web document Framework [Y. Zhu, A. Srikanth, J. Leng, and V. Reddi(2012).] comprises of two components such as page clipping for small-screen devices and page clipping for portal site development. The page clipping for small-screen devices checks the user's HTTP request header. It then identifies the device type and retrieves the HTTP requests from the site. Afterwards it transfers the web page into a tree, adds page-clipping annotations in the tree and uses the annotation-based page-clipping engine to render the page ready for the mobile devices [M. Butkiewicz, H. V. Madhyastha, and V. Sekar(2011)]. The page
clipping for portal site development is a server-side component. The portal creates an annotation-based clipping portal. It combines the dynamic content of the portal and delivers the output of the mobile devices [Y. Chen, X. Xie, W.Y. Ma and H.J. Zhang (2009)].

Xadaptor [Prochkova, V. Singh, and J. Nurminen (2012)] comprises of client database, rule-base and tree processing elements such as content parser, inference engine and content converter. The database keeps the client’s profile. Each client profile maintains the information of a client's preferences and mobile device information. The rule-base maintains various types of rules to adapt the contents. The content parser extracts the original page and breaks the parser into smaller objects. The Inference engine applies the rules from the client database to the objects and the content converter changes the format of the objects [Y. Arase, T. Maekawa, T. Hara, T. Uemukai and S. Nishio (2006)].

The Web Page Tailoring System [Dubey, P. De, K. Dey, S. Mittal, V. Agarwal, (2013)] keeps tracing every movement on the mobile browser. This framework retrieves the user preferences and rearranges the web page layout according to these preferences. The Configure Manager stores the user preferences and the contents get adapted according to the user’s preferences. The Configuration Manager filters out the web page, rearranges the contents and delivers it to the users [Y. Arase, T. Hara, T. Uemukai and S. Nishio (2007)]. Most of the steps for these adaptation frameworks are same, but the techniques are different from each other.

Proposed Work:

The system developer should be able to adapt the content. The below architecture has been developed as a web application. The block diagram explains the various steps involved in the content adaptation, which include RDF Profiles repository, Rule Repository, Rule Engine,Style sheet repository, Page Formatter and web server to retrieve the requested content from the web. The various modules involved in the system and their interaction has been depicted in Fig 1.

The request from the client is sent to the web server and it performs a search to retrieve the information from the internet. The required web page is then sent to the rule engine which finds information applied to the original web page so it can be viewed perfectly using any kind of device. When web content designed to be displayed on desktops is adapted for the small screen size device. The content often looks messy making it difficult for users to read the contents. Hence unwanted information has to be discarded and also certain things like image formats need to be converted from one device may support one format and other may support another. Hence adaptation need to be carried over.

A. Page formatter:

This module deals with managing results and sending them to a page formatter where it sets some basic values like the dimensions of a new image or its values of scale for a runtime adaptation of content to be presented. Thus, this will format the page in accordance with the user device. In this study, jsoup is used for formatting the requested user web page based on the device screen size. The screen size of the device is extracted from RDF repository. Jsoup is a Java library for working with real-world HTML [G. Perrucci, F. Fitzek, G. Sasso, W. Kellerer, and J. Widmer (2009)]. It can parse HTML from a URL, file, or string. It can find and extract data, using DOM traversal or CSS selectors. The HTML elements, attributes and text can be manipulated. It can clean user-submitted content against a safe white list. jsoup is designed to deal with all varieties of HTML found in the wild, from pristine and validating to invalid tag-soup; jsoup can create a sensible parse tree. With the help of this, pictures or images can be formatted easily based on the device screen size.

Fig.1: System architecture.

Algorithm:
Input:
v is the node of the Tree;
Output:
v with modified HTML elements;
Start:
Preprocessing (v)
Iscreen = (DeviceWidth / DeviceHeight) × 200;
If v is decoration or highlight element Then
Remove the element from v and change the background colour properties;
If v is multimedia element and multimedia height, width more than Iscreen Then
MaxHeight = Iscreen;
MaxWidth = Iscreen;
Ratio = height / width;
If height > MaxHeight Then
newheight = MaxHeight;
newwidth = height / Ratio;
Elseif width > MaxWidth Then
newwidth = MaxWidth;
newheight = width × Ratio;
update v with the new width and height;
End

B. Rule engine:
Feature vectors were extracted from the filtered image and fed as inputs to the first layer of neurons of the multilayered neural network. The result obtained from the output layer will be compared to the target output [Rong Pan, Huiqin Wei, Shan Wang (2012)]. Based on this, the network adjusts the weights for each neuron at the output and hidden layer respectively (back-propagation learning method). The network then proceeds to the next image, compares the result with the targeted output and adjusts the weights again until a pre-set goal is met.

The rule engine is an if/then statement interpreter. If/then statements that are interpreted are called rules. The input of this rule engine will be rule execution set and some data output will be selection of rules that needs to be implemented.

Algorithm:
Step 1: Get the rule service provider from the provider manager and get the rule administrator.
Step 2: Get an input stream to a test XML rule set and parse the rule set from the XML document
Step 3: Register the rule execution set and ret a rule runtime and invoke the rule engine
Step 4: Create a stateful rule session and the rule to it.
Step 5: Then Fire all rule that is executing the rules
Step 6: If matches with the rule in a rule repository, then accordingly apply the style sheet

C. Comparison of proposed work with existing experimental frameworks:
A rule engine in tandem with rule base has been used in the system to dynamically carry over the content adaptation policy based on the device. By having a general base for the characteristics of all the communication devices in the rule base, the effectiveness of the adaptation has been increased manifold due to the customized policies [Albasir, A (2013)]. The rule base can also be equipped to hold user based adaptation policies based on their own preferences. This greatly helps the user in obtaining the required content in their most desirable area of interest and in the preferable format [Y.P. Xiao, Z.C. Wang and Q. Li, (2012)]. The style sheet repository helps to reproduce the requested content in the most suitable presentation format according to the communication device. The user experience is greatly enhanced by implementing this presentation layer adaptation over the requested content.

There are several significant differences between the proposed web content adaptation techniques and other existing content adaptation techniques. Therefore, the Xadapto [G. Fox, (2011).], Webpage tailoring system [Y. Arase, T. Maekawa, T. Hara, T. Uemukai and S. Nishio (2006)] and CMo [Y. Arase, T. Hara, T. Uemukai and S. Nishio (2009)] frameworks have been selected for comparison purpose. The proposed work consists of six components. Table 1, shows the comparison of frameworks’ components.

D. Time complexity comparisons:
Block identification algorithm complexity in proposed is O (n) because it uses an iterative loop and recursive function where n is the number of edges in the html tree. The Find block algorithm uses iterative loops and the complexity is O (n). Where n is the number of nodes in the frame tree. Secondly, the proposed object's
identification algorithm complexity is $O(n)$ because it uses iterative loops and recursive function, where $n$ is the vertices and $e$ is the number of edges in the tree.

**Table 1:** Comparison of framework components.

<table>
<thead>
<tr>
<th>Components Used</th>
<th>Xadaptor System</th>
<th>CMo System</th>
<th>Web Page Tailoring System</th>
<th>Proposed Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>User preferences</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Representation of HTML page as a tree</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Block identification</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Object Identification</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Modification object elements</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Toggle the contents for display and hiding</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Table 2:** The time complexity comparison of the algorithms.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Time Complexity</th>
<th>Proposed</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>Block Identification</td>
<td>$O(n)$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Object identification</td>
<td>$O(n+e)$</td>
<td>$O(n^2)$</td>
<td></td>
</tr>
<tr>
<td>Contents height and width calculation algorithm</td>
<td>$O(n)$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Conclusion:**

The main objective of this research is to propose new algorithms for adapting the layout of web content for mobile devices. To fulfill the objectives of the new solutions have been proposed to improve the efficiency and effectiveness for mobile web browsing. The proposed method is an automated mobile web content adaptation system. This increase the usability and friendliness of the webpage for the handheld devices.

However, with the wide scope of future research there are possibilities of overcoming the limitations and of leading the study in many directions. Firstly, the adaptation mechanism should be extended to the tablet PC and other small screen mobile devices, secondly there are some web pages which don't have proper html structures, a mechanism could be built to adapt such unorganized web pages. Finally, there are different sizes of mobile phone screens available and the fuzzy rules mechanism could be used to adapt the contents for these different sizes of screens.

**REFERENCES**


