World Applied Sciences Journal 10 (Special Issue of Computer & Electrical Engineering): 49-58, 2010
ISSN 1818-4952
© IDOSI Publications, 2010

Web Transcoding for Mobile Devices Using a Tag-Based Technique

[1]Mehregan Mahdavi, [2]Hamid Khordadi and [3]Mohammad-Hassan Khoobkar

[1]Department of Computer Science and Engineering, University of Guilan, Iran
[2]Department of Information Technology, University of Shiraz, Iran
[3]Faculty of Computer Science and Engineering, Islamic Azad University, Lahijan Branch, Iran

Abstract: Requirements of users for wireless devices such as laptops and mobiles are different from their needs for wired devices. Wireless networks have limitations such as bandwidth, processing power of devices and size of display screen. Therefore, some techniques are required to enable loading and showing web pages appropriately on mobile devices based on available network bandwidth, processing power of devices and size of display screen. This paper proposes a tag-based technique for transforming web pages into appropriate formats for being sent to and viewed on mobile devices. This technique enables generating different versions of web pages for different mobile devices. Our experiments show that our proposed method can decrease loading time of web pages significantly. Moreover, it can reduce network bandwidth consumption by using appropriate tags by content providers in order to include only essential parts of web pages and/or include smaller size images, thus reducing network bandwidth consumption. Inserting such tags within web pages by web site designers can result in producing different versions of web pages for different mobile devices.

Key words: Transcoding · Server-based · User-based · Proxy-based · Tags · Loading time · Network bandwidth · Mobile

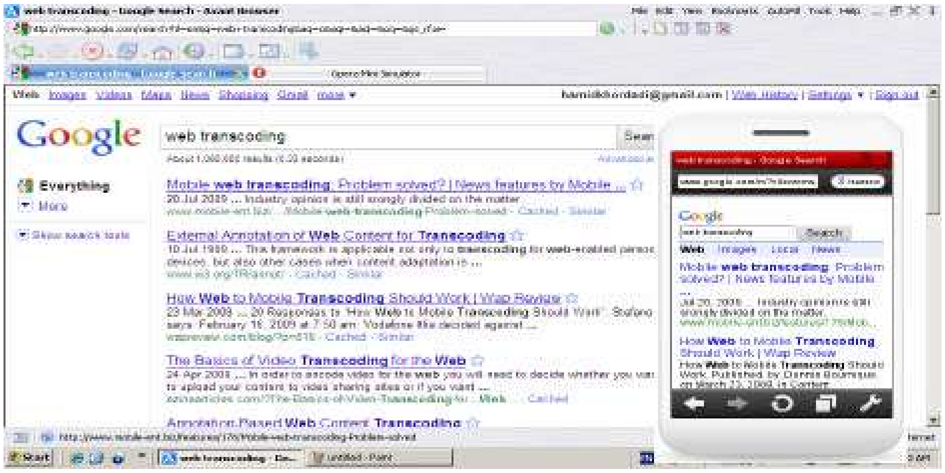
INTRODUCTION

Accessing the Internet using mobile devices is common these days. These devices have already changed the way that businesses work and have established new jobs. It has also changed the way that individuals use the Internet for accessing information, learning purposes, shopping, gaming, etc. In fact, such devices enable accessing web contents and services virtually anywhere and anytime. These may include text, image, voice, video, programs and geographical situation of an individual. Most of mobile users access such contents and services. Unfortunately, they are designed for PCs with large monitors (compared to mobile devices) and high bandwidth Internet access. Current research indicates that%30 of mobile web users are not fully satisfied with their experience for Internet access. Most of PCs have 1024*768 or more screen resolution, while the screen resolution of Personal Digital Assistants (PDA) is around 240*320. Mobile phones provide a lower screen resolution of around 176*208. Although, Mobile devices are different in terms of network type and available resources (such as memory and saving capability), size of display screen and format of data, but they can normally run similar applications. Therefore, it is necessary to provide methods and techniques for showing information properly regardless of the environment and display screen features [1, 2].

Transcoding means a process for choosing, changing or producing the content (text, image, voice and video) to be conformed with the current environment that user prefers. This can be achieved by changing the size of an image, its resolution and quality, replacing voice by its equivalent text or video by a number of images [3, 4]. Fig 1 shows an example. In this figure, the results of a Google search are shown and compared in a browser of a PC and of a PDA.

There are a number of important parameters that are the main reasons for Transcoding. They include making web pages more user-friendly and show them appropriately in different mobile devices with different screen sizes, reducing size of web pages in order to reduce network bandwidth and decreasing their loading time. These parameters could be used in order to compare and evaluate different Transcoding systems. Current Transcoding systems achieve these goals by either

Corresponding Author: Mehregan Mahdavi, Department of Computer Science and Engineering, University of Guilan,
Rasht, Iran. Tel: 00989126798967, Fax: 00981316690271, E-mail: mehregan.mahdavi@gmail.com


web transcoding - Google Search - Avant Browser
Opera Mini Simulator
Web Images Videos Maps News Shopping Gmail more ▾
hamidkhordadi@gmail.com | Web History | Settings ▾ | Sign out
Google
web transcoding
Search
Everything
More
Show search tools
web transcoding - Google Search
Web Images Local News
Start
Untitled - Paint

About 1,060,000 results (0.20 seconds)

Mobile web transcoding: Problem solved? | News features by Mobile ...
20 Jul 2009 ... Industry opinion is still strongly divided on the matter
www.mobile-ent.biz/.../Mobile-web-transcoding-Problem-solved - Cached - Similar

External Annotation of Web Content for Transcoding
10 Jul 1999 ... This framework is applicable not only to transcoding for web-enabled personal devices, but also other cases when content adaptation is ...
www.w3.org/1999/... - Cached - Similar

How Web to Mobile Transcoding Should Work | Wap Review
23 Mar 2008 ... 20 Responses to "How Web to Mobile Transcoding Should Work". Stefano says: February 16, 2009 at 7:50 am. Vodafone the decided against ...
wapreview.com/blog/?p=615 - Cached - Similar

The Basics of Video Transcoding for the Web
24 Apr 2008 ... In order to encode video for the web you will need to decide whether you want to upload your content to video sharing sites or if you want ...
ezinearticles.com/?The-Basics-of-Video-Transcoding-for-...-Web... - Cached

Annotation-Based Web Content Transcoding

Mobile web transcoding: Problem solved? | News features by Mobile ...
Jul 20, 2009 ... Industry opinions still strongly divided on the matter.

How Web to Mobile Transcoding Should Work | Wap Review
How Web to Mobile Transcoding Should Work. Published by Dennis Bournique on March 23, 2008, in Content

Fig. 1: Comparison of a Google search in a PC and a mobile device

removing non-vital parts of web pages such as pictures or advertisements, or moving some content from one page to a new page and creating a link in the initial page to the new page in order to reduce size and dimensions of web pages.

In this article we propose an automatic Transcoding system. This is achieved by inserting special Transcoding tags that will be inserted within current web pages by a site administrator. These tags enable automatic Transcoding of web pages according to the site administrator's priorities. We have used a web simulating environment (WAP Proof 2008) to evaluate our proposed system[1]. We have also simulated this system using a number of mobile devices. Our experiments indicate that our proposed system performs effectively in terms of reducing network bandwidth, reducing loading time and satisfying users and site designer.

This paper is organized as follows: The next section gives an overview of previous research and existing tools for Transcoding. Our proposed method is explained in the following section. Experimental results are then presented. Finally, the paper is summarized and concluded.

RELATED WORK

Various mechanisms exist for Transcoding web pages. In general, these mechanisms can be divided into three categories: Client-based, Proxy-based and Server-based.

In client-based techniques, mobile devices receive the whole page from HTTP server and transform the content to the required format on the mobile device as shown in Figure 2. The main advantage of this technique is that nothing needs to change in web server. For example, Opera Mini[2] enables client-based Transcoding which is called Small-Screen Rendering (SSR) in mobile browsers. This technology changes size of web pages appropriately for mobile phone and provides vertical and horizontal movement in the web pages. Since, in this method, the Transcoding is implemented on mobile devices, only a limited number of mobile devices are capable of using this technology. Moreover, the resulted pages might have a poor quality, mostly in terms of user satisfaction [5].

Client-based Transcoding is not very common, as this method needs to use the device itself for transcoding web pages and the mobile devices need to be powerful enough to handle transcoding [6]. Moreover, this method does not reduce network bandwidth usage, as the whole web objects need to be transferred to the mobile device in order for the transcoding process to be done. Cascading Style Sheet (CSS) [7] is a common technique used for client-based Transcoding. Web page designers can use CSS for defining different rules for different media. In this method, presentation is completely separated for the content and different browsers in different devices receive the same content. However, the relevant CSS applies the relevant changes on the contents instantly for presentation purposes. CSS uses XML-based languages, such as Hyper Text Mark-up Language (HTML) and Extensible Hyper Text Mark-up Language (XHTML).

[1] http://www.wap-proof.com

[2] http://www.opera.com/mobile/

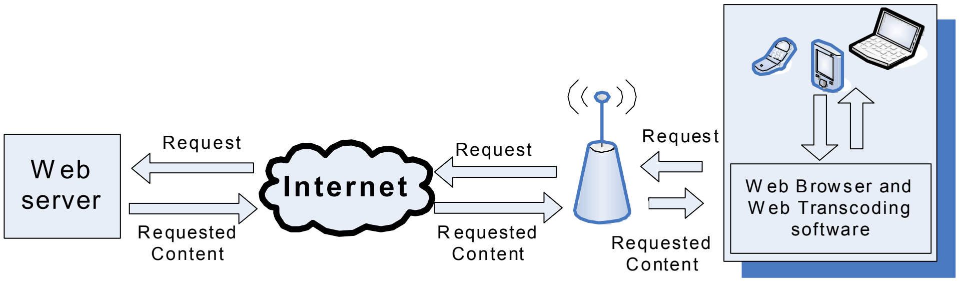


Fig. 2: Client-based Transcoding

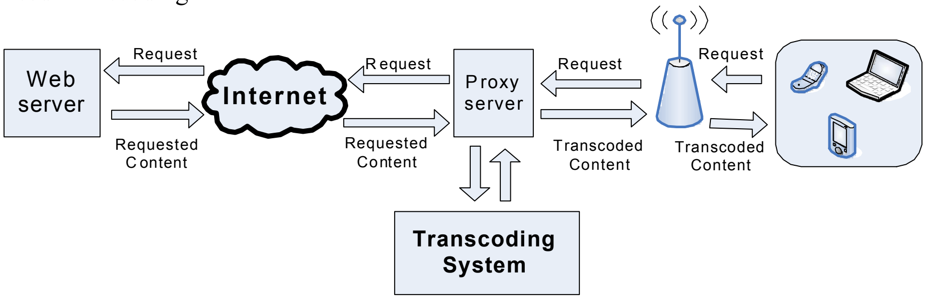


Fig. 3: Proxy-based Transcoding

As an example, Opera uses CSS for rendering web pages based on attributes of the mobile device. As mentioned earlier, users can browse web pages on small screens. They will not need horizontal scroll. Instead, users can vertically scroll web pages. Moreover, the content can be magnified by users.

In proxy-based techniques (which are nowadays used by many commercial web sites) Transcoding process is done by a transparent node called a proxy server. A proxy server is located between client and web server [8] (as shown in Figure 3). The web server transfers web objects to the proxy server and then the proxy server transcodes the content according to type of the device which is mentioned in the user request. The resulted content is then transferred to the user [9, 10].

WebAlchemist [11, 12] is an HTTP proxy for applying advanced transcoding. It consists of four modules: an HTML Tokenizer, a Grammar Corrector, an Internal Representation Generator and a Transcoding Manager. The HTML Tokenizer module converts HTML pages into tokenized strings using HTML tags. The Grammar Corrector module corrects syntactic errors in HTML pages. The Internal Representation Generator receives the tokenized strings, which are the corrected HTML pages and creates their tree-based representation. The Transcoding Manager performs the transcoding process. WebAlchemist[13] was originally designed for handheld devices with browsers that support HTML. It converts the original HTML pages into pages that do not need horizontal scrolling. It also tries to produce pages that fit on small screens by reducing the size of images.

WebViews [14] is a system that enables delivering Web content to wireless devices. When browsing the web on desktop computers, users can record their browsing activities using the WebViews' Recorder applet. Users can select their favorite page fragments (called Web Views) and save them in a database. Their browsing experience is stored in the form of XML files. These created Web Views can be later accessed by the users via their specified URLs on a mobile device. Webviews is based on three components: a server, separate proxies for different markups and various clients.

Web Intermediaries [15] is a Transcoding framework that is based on a programmable proxy server that receives HTTP requests from clients and generates HTTP responses suitable for being displayed on the requesting device. This proxy server is called Web Intermediaries (WBI). It uses a set of plug-ins for this purpose. When a request is received by WBI, it requests the content from the original web server. If there is no annotation with the request, the content is forwarded to the client. Otherwise, it extracts the characteristics of the requesting device from the HTTP header [16] and generates an appropriate version of the page according to the specified annotation. WBI can break the page into sub-pages before sending the result to the client according to the specified annotation.

Power Browser [17, 18] is a web browser designed for mobile devices especially PDAs. It operates based on a proxy server and uses text summarization techniques for representing content on such devices. Each Web page is divided into a number of Semantic Textual Units (STU). Each STU can be: hidden, partially visible, fully visible, or summarized.

In server-based techniques, the content is transformed by the web server and clients receive the Transcoded content on their mobile device. This technique does not have the limitations of client-based technique, e.g., limitation on processing power of the clients' device. There are two mechanisms widely used in such techniques. The content can be either chosen based on the characteristics of the device from a set of different versions that have already been generated, or it can be dynamically generated [6]. Different systems might produce different content with different layout and browsing capabilities.

There are a number of Java-based technologies that are commonly used in server-based techniques. Jave Servlets, Java server Pages (JSP), JSP Tag Libraries, Java Beans and Enterprise Java Beans (EJB) are the most common ones. Java Servlet is a hardware-independent Java technology for generating dynamic web-based applications. A Servlet is a Java Class that is translated to byte code and can access a rich set of HTTP Application Program Interface (API), using a web server such as Tomcat and JBoss [19]. JSP is based on Servlet technology and helps in developing dynamic web content more easily. A JSP can contain HTML tags, Wireless Mark-up Language (WML) tags, or JSP tags that enable web servers generate dynamic content [20,21]. Java Beans are Java Classes that conform to certain coding standards and can be used as software components and application programs. They are considered as reusable software units with a uniform interface that encapsulate data components and hide implementation details [22]. EJBs are Java-based software components that can be developed and used in multi-layered environments for operations such as database access or running the business process [23].

There are other server-based architectures based on eXtensible Mark-up Language (XML) and Extensible Stylesheet Language Transformations (XSLT) in which tags are defined for mobile devices using XML tags. The content is stored in XML files and compatibility with mobile devices is achieved using XSLT files [24]. Apache Cocoon [25] is an Open Source Architecture that was originally introduced as a Java servlet provided content in several formats and languages such as HTML, eXtensible Hyper Text Mark-up Language (XHTML), XML, WML and Scalable Vector Graphics (SVG).

User Interface Description Language (UIDL) is another server-based technique [26]. User Interface Mark-up Language (UIML) is an XML-based language that can be used for sending content to different devices in a device-independent fashion. eXtensible Interface Mark-up Language (XIML) is a descriptive language for providing different user interfaces [28]. Renderer Independent Markup Language (RIML) is a language aimed at enabling development of highly usable mobile applications [29]. Model-View-Controller (MVC) is a design pattern that consists of three separate components: model, view and controller. The model is responsible for keeping data and state of application programs. The view is a user interface that provides users with the information about the model. The controller interacts with the user, accepts input from the user and impacts the model. Struts is an example of an Open Source framework based on MVC [30]. Java Server Faces (JSF) provides a Web application development framework which supports MVC and enables developing web applications for heterogeneous devices on the web [31].

PROPOSED TRANSCODING SYSTEM

Existing Transcoding systems use algorithms and techniques in order to generate Web pages according to the characteristics of mobile devices. However, the generated content might not satisfy users in terms of readability and user-friendliness. They might not be also able to generate appropriate results for all mobile devices based on content providers' requirements. For example, in a server-based Transcoding system, the web site designer can manually design different versions of each web page for each mobile device. Our proposed tag-based Transcoding system is a server-based Transcoding system that enables content providers to automatically generate content appropriate for different mobile devices. In this system a set of tags are defined, each of which provides a task to be performed when generating the content. By including appropriate tags within web pages, web site designers enable automatic Transcoding by the system. These tags will be explained later in this section. The general architecture of the system is shown in Figure 5. When a request is received, the Transcoding system checks if the request comes from a normal PC or a mobile device.

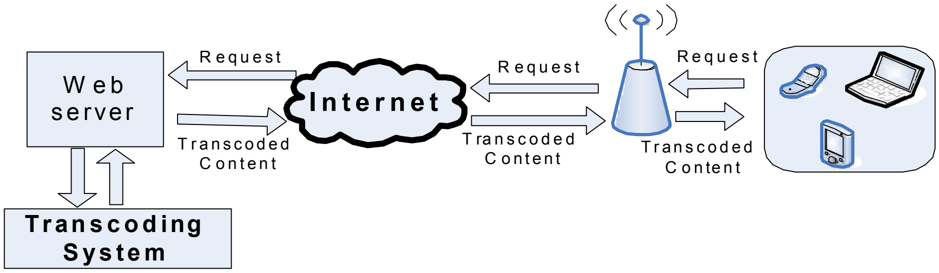


Fig. 4: Server-based Transcoding

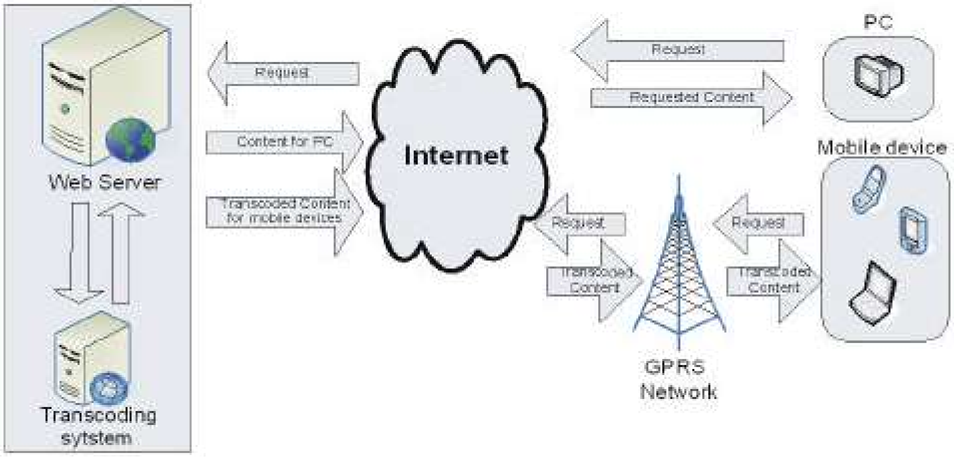


Fig. 5: The Proposed Transcoding System

If the request comes from a normal PC, it sends the content to the requester. Otherwise, it applies the tags which have been already included within the requested web page, in order to generate the appropriate version of the content for the requesting mobile device. The results are then sent to the requester.

As mentioned earlier, Transcoding systems enable appropriate presentation of web pages in different mobile devices. Moreover, they reduce page loading time and also network bandwidth consumption. These are achieved by: eliminating non-essential parts of web pages for decreasing their size, thus reducing network bandwidth consumption and faster loading time. These parts can include images or advertisements, flash files and music in web pages.

Moving some parts of the content from web pages to a new page and inserting a link to the new page in the original page instead. This can help reducing page size and dimensions and therefore enhancing presentation of web pages and reducing their loading time.

Most of existing Transcoding systems address the above-mentioned items. However, the main issue is to generating different versions when a web page is updated. This has to be applied manually by the site designer to all the different versions of the web page for different mobile devices. The proposed Transcoding system provides an automatic solution when the content changes.

In the rest of this section we explain the proposed Transcoding system. We first introduce the defined "Tag Library" in this system. We then explain the "Transpreter" (stands for Transcoding Interpreter). It reads web pages that include Transcoding tags and generates appropriate version of web pages by applying the included tags.

The Tag Library: We have designed a tag library for our Transcoding system. These tags are compatible with HTML. In order to distinguish these tags from other HTML tags, they are all started with a # symbol. As a result the system does not interfere with HTML content and if these web pages are uploaded on a web server without our Transcoding system, these tags are easily ignored by the system and will not have any effect on the system. The general structure of these tags is as follows:

```
<! -- /#tagname:tagvalue --> content <!-- #/ -->
```

In this structure, <!-- /#tagname:tagvalue -->, is the beginning of the tag that is going to be applied on "content" and <!-- #/-- > is the end of the tag.

The Tag Library that was designed and applied in the proposed Transcoding system includes the following tags:

Visible: False: This tag is used to eliminate parts of web page that are not required to be shown after Transcoding. For example the following deletes an image named "top.gif" along with its containing table from the transcoded page. It should be noted that this tag will be ignored by web browser if the transcoding system is not applied on the web server, as it will be considered as a comment by the web browser.

```
<! -- /#visible: false -->
<table> <tr> <td> <img src="top.gif" /> </td> </tr>
</table>
<!-- #/ -->
```

Link:exp: This tag, after Transcoding, moves the content to a new page and instead creates a link to the new page named "exp" in the current web pages. For instance, the following expression moves the "contents of news" along with its containing table to a new page and replaces that with a link named "news" in the current page. An example of using this tag in a web page is shown in Figure.

```
<!-- /#link: news -->
<table> <tr> <td> … contents of news… </td> </tr>
</table>
<!-- #/ -->
```

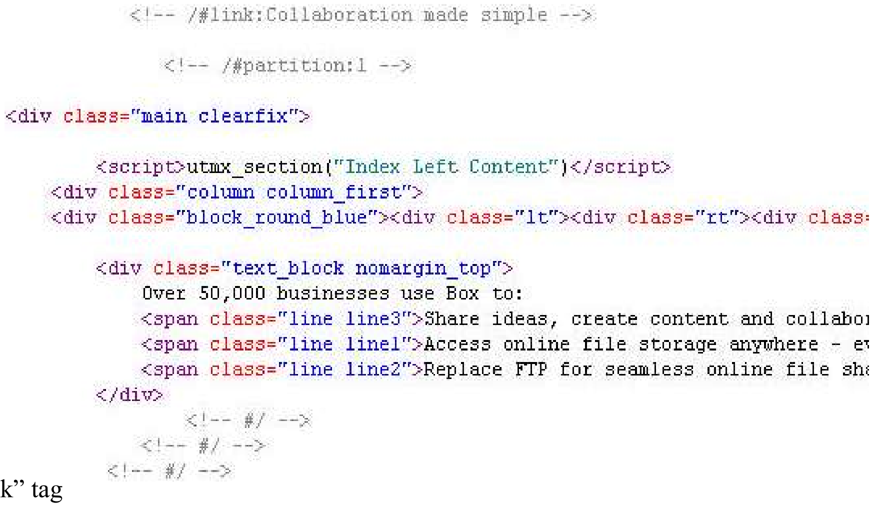
Partition: Num: As is shown in Figure 7, with this tag, we can split web page into different partitions and specify priority for each partition ("num"). The priority is a number which begins with one for the highest priority. The purpose of this tag is defining the order of partitions in the Transcoded page. For example, as can be seen from the following expression, "content 2" appears before "content 1" in the normal situation. However, after Transcoding, "content 1" is shown first, followed by "content2".

```
<!-- /#partition:2 -->
<table> <tr> <td> content 2 </td> </tr> </table>
 <!-- #/ -->
<!-- /#partition:1 -->
 <table> <tr> <td> content1 </td> </tr> </table>
 <!-- #/ -->
```

In our Transcoding system, each web page consists of at least one partition. Other tags such as "visible" and "link" will be included in "partition" tag. The Transcoding system is applied only on body of web pages and there will be no change in head of web pages. As a result, included codes such as JavaScript and CSS can be used without any problem. Figure 8 shows the results based on the tags that are partly shown in Figure 6 and Figure 7.

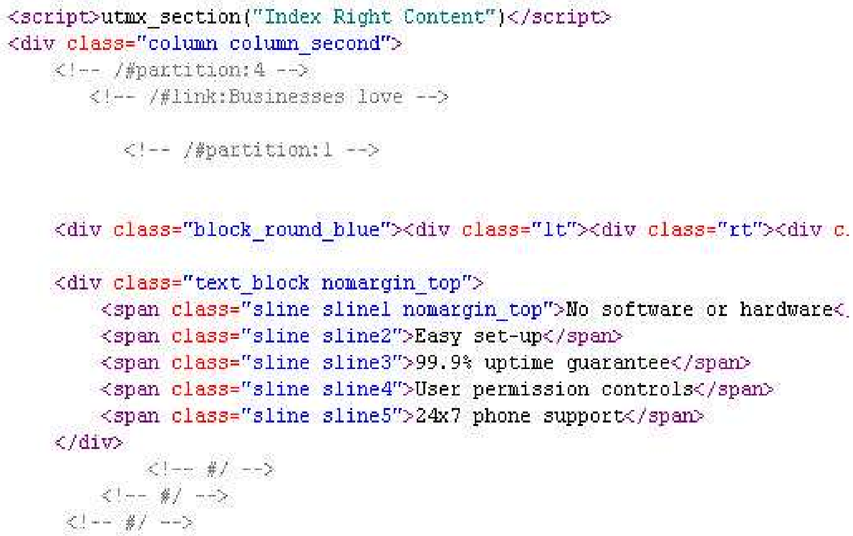
The Transpreter: In order to interpret the tags within web pages and generate the output, an interpreter was designed and implemented. We call it "Transpreter", which stands for Transcoding Interpreter. It has been implemented as an.EXE or.DLL file (i.e., Transcodingdll.EXE or Transcodingdll.DLL). It can be installed and ran on the web server (e.g., Microsoft's IIS). It intercepts the requests to the web server and performs Transcoding according the tags within the requested web pages. The Transpreter was implemented in C# programming language which enables writing programs in.Net framework. The algorithm used by the Transpreter is depicted in Figure 9. Upon receiving a request, it identifies type and characteristics of the requesting device. This information is included in the HTTP request. If it is a mobile device, our Transcoding system will be executed and the generated output will be sent to the user. If the request does not come from a mobile device, the web page will be sent to the user without any changes.

In order to explain how the output is generated for mobile devices, we provide an example as shown in Figure 10. Figure 10 (a) shows a web page (www.box.com) which is related to sharing information among users. We have included our Transcoding tags within the Web page and shown the result using Internet Explorer (IE) on a PC. Its size is about 206 KB. As can be seen from the figure, these tags do not have any effect on the appearance of the web page. Figure 10 (b) shows the results on a mobile device after applying Transcoding tags on the content by splitting the page into several sub-pages and inserting a link for each sub-page in the first page. The size of the first page is about 4 KB and its loading time is about 6 seconds.



```
          <!-- /#link:Collaboration made simple -->

             <!-- /#partition:1 -->

<div class="main clearfix">

        <script>utmx_section("Index Left Content")</script>
    <div class="column column_first">
    <div class="block_round_blue"><div class="lt"><div class="rt"><div class=

        <div class="text_block nomargin_top">
            Over 50,000 businesses use Box to:
            <span class="line line3">Share ideas, create content and collabo
            <span class="line line1">Access online file storage anywhere - e
            <span class="line line2">Replace FTP for seamless online file sh
        </div>
                <!-- #/ -->
            <!-- #/ -->
         <!-- #/ -->
```

Fig. 6: Using “link” tag



```
<script>utmx_section("Index Right Content")</script>
<div class="column column_second">
    <!-- /#partition:4 -->
       <!-- /#link:Businesses love -->

          <!-- /#partition:1 -->


    <div class="block_round_blue"><div class="lt"><div class="rt"><div c

    <div class="text_block nomargin_top">
        <span class="sline sline1 nomargin_top">No software or hardware<
        <span class="sline sline2">Easy set-up</span>
        <span class="sline sline3">99.9% uptime guarantee</span>
        <span class="sline sline4">User permission controls</span>
        <span class="sline sline5">24x7 phone support</span>
    </div>
            <!-- #/ -->
        <!-- #/ -->
     <!-- #/ -->
```

Fig. 7: Using “partition” tag

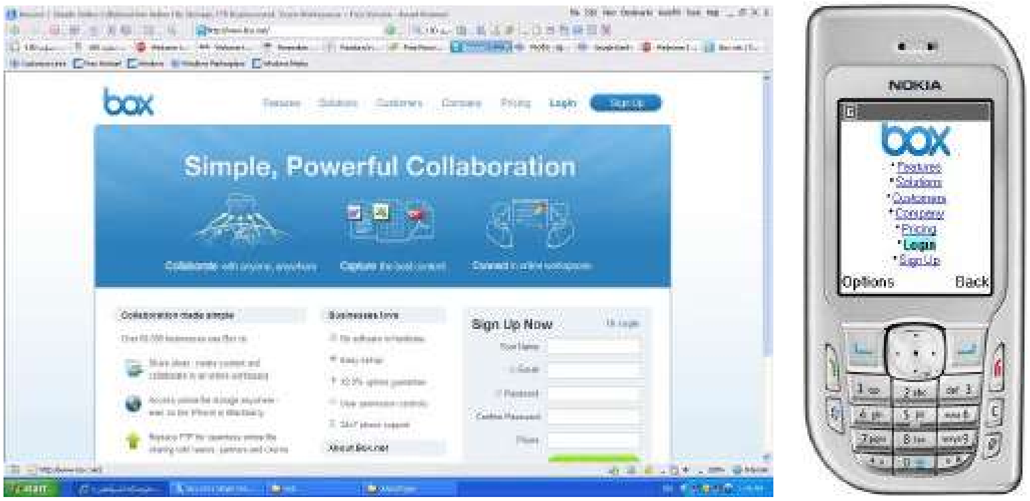


Fig. 8: Results of tag-based Transcoding

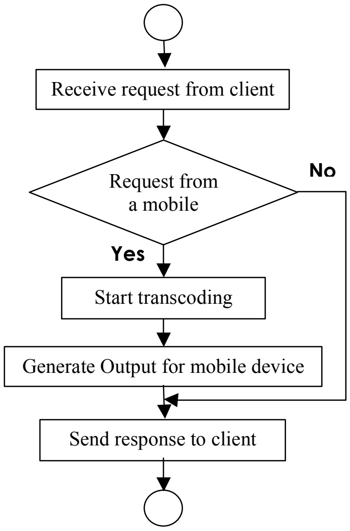


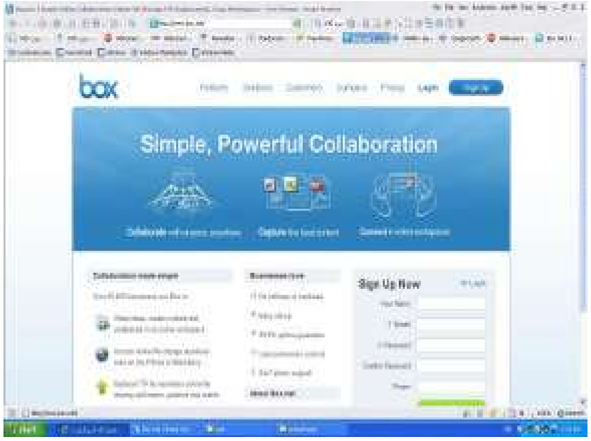
Fig. 9: The algorithm used by the Transpreter

EXPERIMENTAL RESULTS

In order to evaluate our proposed system and its performance, we have used WAP Proof 2008 software on a PC with a 2.2 GHZ Pentium IV processor with a 128Kb/s Internet and compare the performance results before and after Transcoding. We have used Page size and loading time as the major parameters for the comparison.

We have examined several popular web sites and included some Transcoding tags within the web pages. Then we compared the size and loading time of the original pages with those of Transcoded pages using our proposed system. The results are presented in Figure 12 and Figure 13.

As can be seen in Figure 12, page size can be reduced significantly by our Transcoding system. In existing Web sites, the whole page has to be loaded by the user, even



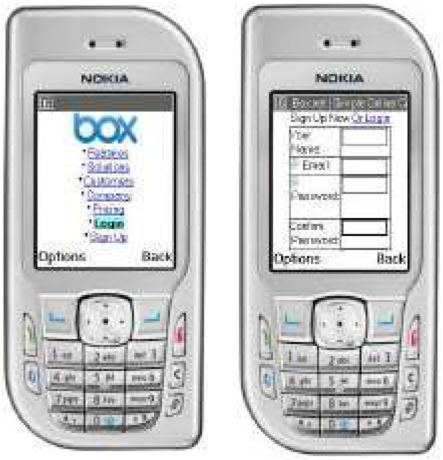
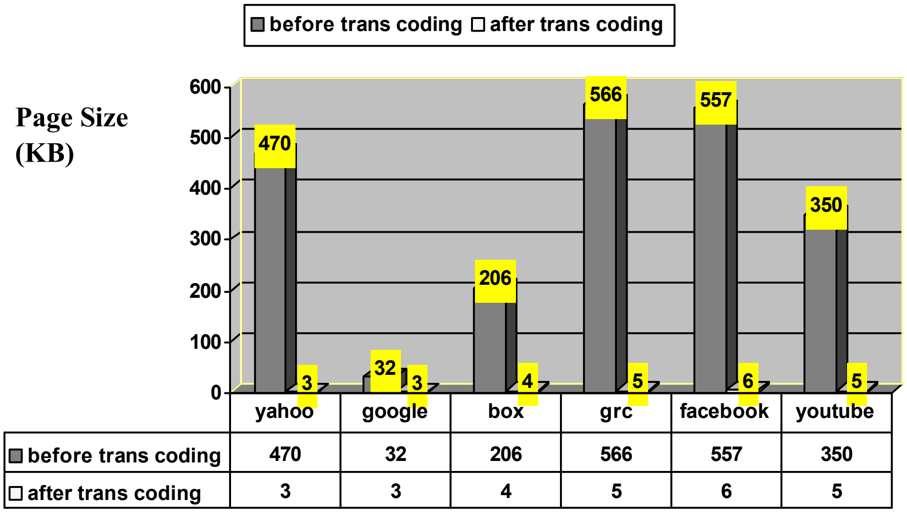
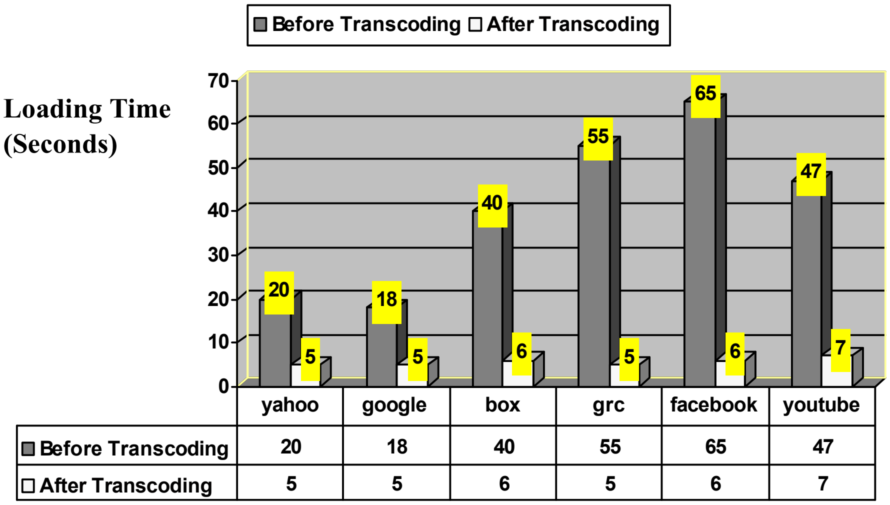


Fig. 10: Transcoding a web page



	yahoo	google	box	grc	facebook	youtube
▨ before trans coding	470	32	206	566	557	350
□ after trans coding	3	3	4	5	6	5

Fig. 12: Comparison of page size before and after Transcoding



	yahoo	google	box	grc	facebook	youtube
▨ Before Transcoding	20	18	40	55	65	47
□ After Transcoding	5	5	6	5	6	7

Fig. 13: Comparison of page loading time before and after Transcoding

though some parts may not be necessary. Therefore, size of such pages might be large. However, our proposed Transcoding system results in reducing the size by either eliminating unnecessary parts defined by the web site designer, or splitting each page into several pages. The results of page loading time are shown in Figure 13. The results show that our Transcoding system can reduce page loading time significantly. This may result in better user satisfaction as well.

CONCLUSION

In this paper, we have examined existing solutions for Transcoding web pages for mobile devices. We have addressed the issue of generating different versions of web pages when they are modified, by proposing a tag-based Transcoding system. It automatically generates the appropriate version of a web page to be displayed on a specified mobile device. This is achieved by including some Transcoding tags to be inserted within web pages by web site designers. Our experiments show that our proposed method can provide better user satisfaction while decreasing page size and page loading time significantly.

REFERENCES

1. MDA, Q3 2006. Mobile Internet Figures Continue To Grow, 2007. http://www.themda.org/PressReleases/ Page_Press_PressReleases_LatestStats.asp.
2. OPA, Going Mobile: An International Study of Content Use and Advertising on the Mobile Web," 2007. http:// www.online-publishers.org./ media/ 176_W_opa_going_mobile_report_mar07.pdf.
3. Lewis, R., 2003. Glossary of Terms for Device Independence, W3C Working Draft, http:// www.w3.org/ TR/di-gloss.
4. Canali, C., V. Cardellini and R. Lancellotti, 2003. Squid-based proxy server for content adaptation, *Technical Report TR-2003-03*, Dept. of Computer Engineering, Univ. of Roma.
5. The Web adapted to every screen size: Opera solves the problem of rendering Web content on medium-sized screens" Opera press release, 2004. http:// www.opera.com/ pressreleases/ en/2004/01/20/.
6. W3C: Authoring Techniques for Device Independence, 2004. http://www.w3.org/TR/di-atdi/.
7. W3C: Cascading Style Sheets Level 2 (CSS 2.1), 2006. http://www.w3.org/TR/CSS21.
8. Zhang, D., 2007. Web Content Adaptation for Mobile Handheld Devices, Communications of the ACM (CACM), 50(2): 75-79.
9. Colajanni, M. and R. Lancellotti, 2004. System Architectures for Web Content Adaptation Services, In IEEE Distributed Systems online, 5: 5.
10. Laakko, T. and T. Hiltunen, 2005. Adapting Web Content to Mobile User Agents, In IEEE Internet Computing, IEEE Communications Soci., 9(2): 46-53.
11. Hwang, Y. E. Seo and J. Kim, 2002. WebAlchemist: A Structure-Aware Web Transcoding System for Mobile Devices, in Proceedings of the Mobile Search Workshop, Honolulu.
12. Hwang, Y., E. Seo and J. Kim, 2003. Structure-Aware Web Transcoding for Mobile Devices, in IEEE Internet Computing, 7(5): 14-21.

13. Whang, Y.H., C.H. Jung, J.H. Kim and S.K. Chung, 2001. WebAlchemist: A Web Trtanscoding System for Mobile Web Access in Handheld Devices, SPIE ITCOM.
14. Freire, J., B. Kumar and D. Lieuwen, 2001. WebViews: Accessing Personalized Web Content and Services, in Proceedings of the 10th World Wide Web Conference, Hong Kong, pp: 576-586.
15. Hori, M., G. Kondoh, K. Ono, S. Ichi Hirose and S. Sing, 2000. Annotation-Based Web Content Transcoding, in Proceedings of the 9th International World Wide Web Conference on Computer Networks, Amsterdam, pp: 197-211.
16. Nielsen, H., P. Leach and S. Lawrence, 2000. An HTTP Extension Framework, RFC 2774.
17. Buyukkokten, O., H. Garcia-Molina and A. Paepcke, 2001. Seeing the Whole in Parts: Text Summarization for Web Browsing on Handheld Devices, in Proceedings of the 10th International World Wide Web Conference, Hong Kong, pp: 652-662.
18. Buyukkokten, O., A. Paepcke and H. Garcia-Molina, 2001. Text Summarization of Web Pages on Handheld Devices," in Proceedings of the Workshop on Automatic Summarization (NAACL 2001), Pittsburgh.
19. Sun Microsystems: Java Specification Request 154: Java Servlet 2.4 Specification, 2004. http:// jcp.org/ en/ jsr/ detail?id=154.
20. Fields, D., M. Kolb and S. Bayern, 2002. Web Development with Java Server Pages, CT: Manning Publications Co., Greenwich.
21. Sun Microsystems: Java Specification Request 245: Java Server Pages 2.1, 2005. http://jcp.org/ aboutJava/ communityprocess/ edr/jsr245/index.html.
22. Crnkovic, I. and M. Larsson, 2002. Building Reliable Component-Based Software Systems, Artech House Inc., Norwood, MA.
23. Roman, E., S. Ambler, T. Jewell and F. Marinescu, 2005. Mastering Enterprise JavaBeans," 3rd Edition, Wiley Publishing, Indianapolis.
24. Biancheri, C., J. Christophe, R. Pazzaglia and G. Paddeu, 2001. EIHA: Deploying Web and WAP Services Using XML Technology, in SIGMOD Record, 30(1): 5-12.
25. Moczar, L. and J. Aston, 2002. Cocoon Developer's Handbook, SAMS Publishing, Indianapolis.
26. Szekely, P., 1996. Retrospective and Challenges for Model-Based Interface Development, in Proceedings of the 3rd International Eurographics Workshop on Design, Specification and Verification of Interactive Systems, Namur, pp: 1-27.
27. Anegg, H., G. Niklfeld, M. Pucher, R. Schatz, R. Simon, F. Wegscheider, T. Dangl and M. Jank, 2004. Multimodal Interfaces in Mobile Devices-The Mona Project, In Proceedings of the workshop on emerging applications for wireless and mobile access, NY.
28. Puerta, A. and J. Eisenstein, 2002. XIML: A Common Representation for Interaction Data, In Proceedings of the ACM Conference on Intelligent User Interfaces (IUI '02), San Francisco, pp: 214-215.
29. Consensus: Adaptation Engine: Installation and Configuration Guide, 2003.http://www.consensus-online.org/ publicdocs/ D7public.pdf.
30. Apache Software Foundation: Tomcat Web Container, 2005. http:// jakarta.apache.org/ tomcat/ index.html.
31. Sun Microsystems: Java Specification Request 252: JavaServer Faces 1.2, 2004. http:// jcp.org/ en/ jsr/ detail?id=252.