



ISSN 2047-3338

Mark up Based Content Adaptation Technique

Susmita Ghosh Chattopadhyay¹ and Arunasish Acharya^{2,*}

^{1,2}School of Educational Technology, Jadavpur University, India

¹sapta.mita@gmail.com, ²arunasisha@yahoo.com

Abstract– Accessing the Web nowadays in mobile phones has become common and popular worldwide. While mobile phones are becoming more popular, wireless communication vendors and device manufacturers are seeking new applications for their products. Most available Web pages are designed for desktop PC and are not suitable for displaying or browsing on mobile phones because of its small display and limited memory, processing power. In this paper, a content adaptation technique is proposed where adaptation is done using predefined HTML tag sets. All the contents in different format satisfying different device specifications are stored in a single html program. After detection of device, the required portion of the program for the requested content is delivered and the client's browser renders the web page.

Index Terms– Device Detection, Content Adaptation and Markup

I. INTRODUCTION

THERE is a proliferation of using different handheld devices to access internet. This is due to advances in computing and communication technologies have allowed wide range of devices, from workstation to cellular phones, to connect Internet and access Web contents. For this reason the web server has to provide heterogeneous form of same web content according to the capability of the handheld devices e.g. mobile phones due to the following limitations in these devices- display resolution, memory space, CPU power, and network bandwidth as well as software running on it. Most of the web contents are generally designed for desktop. Content adaptation is the process to personalize these web contents to meet client's preferences.

II. NOTION OF CONTENT ADAPTATION

Content adaptation is the process to personalize i.e., to generate, mutate any web content according to the specifications in the client profile. The client or the device requests any web page to the server. The server detects the device and finds its limitations from the profile. After downloading the web page or taking snapshot of the web page, the server either sends directly to the device (if the client is a PC or a laptop) or modifies the web contents in the page

(e.g., takes all the text in a paragraph, reduces the resolution and size of the image etc) and redirects to the handheld device like mobile, PDA etc.

III. DIFFERENT APPROACHES FOR ADAPTATION

Adaptation can occur on the client side, server side or in an intermediate one between these two namely proxy server.

A. Client Side Approach

In this approach the same content is delivered to all the devices rather clients. The client then personalizes the content i.e. change the format or size as required according to the client's specification. But the limitation in this approach is the client's lack of computing power and bandwidth. For this reason the transcoding process of client side approach is very limited or impossible for the multimedia content because these kinds of contents need very expensive transcoding process. To overcome this limitation server side approach is introduced.

B. Server Side Approach

In server side approach, the content adaptation system is merged with the server. Multiple forms of same content are stored in the server. Wireless applications depend on device-specific features such as input/output mechanisms, screen sizes, computing resources, and support for various multimedia formats and languages. This leads to the need for multi-source authoring - the creation of separate presentations for each device type or, at least, for each class of devices. After getting the request the server detects the client and sends the corresponding adapted content. Since already adapted contents get delivered the transmission time gets reduced. To get device independent format of content, the content is represented in XML (Extensible Mark Up Language). XSLT is used to dynamically adjust web content to various devices. General Web architecture will be looked like as Fig. 1.

C. Proxy-based Approach

In this approach the contents are adapted in the fly before sending to the client. The proxy server can store the transcoded form of the content for further use. Thus reduces the computational load of the server. Both the proxy server

Web Contents in Web Page

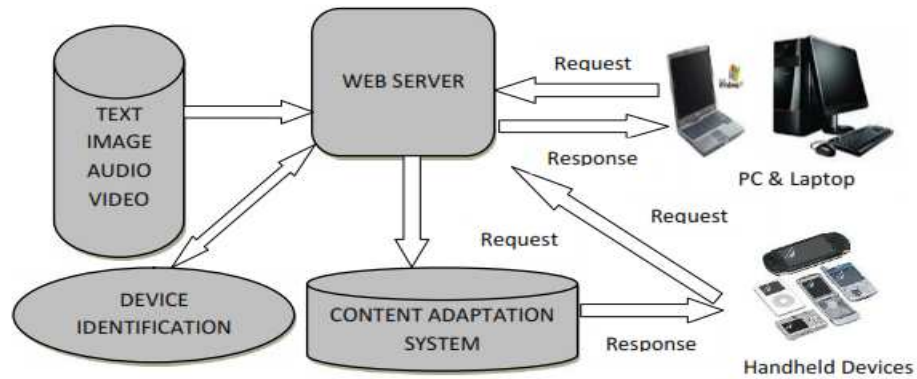


Fig. 1. Content Adaptation Process

and the content server need to know the client to send the appropriate content.

IV. PREVIOUS WORKS

Davide Bellinzona, Patrick Vitali present adaptation of contents designed for desktop for mobile devices using tag libraries [1]. They proposed a framework called Alembik [7] for the transcoding of multimedia content. Alembik is based on serverside solution. The objective is to define tags and their runtime utilization for multimedia content adaptation. Alembik is based on the Transcoding Interface Standard (STI defines the transformations and functionalities which should be implemented by the serverside in a content adaptation system) defined by Open Mobile Alliance (OMA). Three types of contents considered- image, audio & video.

But available tags can be extended for text, presentations.

Mehregan Mahdavi, Hamid Khordadi and Mohammad-Hassan Khoobkar present a content adaptation method for mobile devices using a tag based technique [3]. This is an automated transcoding system that enables generating different versions of web pages for different mobile devices. The main objectives of this paper are to decrease loading time of web pages and reduce network bandwidth consumption by using tags by content providers in order to include only essential part of web page. This technique also contains a transcoding interpreter (transpeter).

Algorithm of this transpeter is given in the following Fig. 2. Niranjan Mangal proposed in his thesis paper how content can be customized and adapted in mobile phone [2]. The objective is to find a better way of navigating and browsing a large web page in mobile device so that the web content found in the desktop can be accessed in the mobile device in the same manner. The aim is to develop a web browser to render Html web pages easily. The rendering technique uses vector class to store the output.

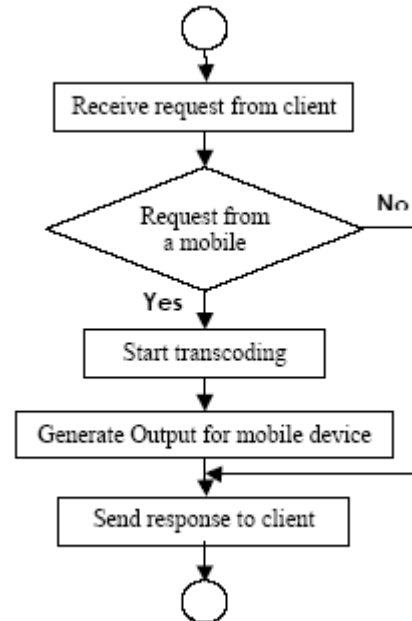


Fig. 2. Algorithm of transpeter [3]

e.g. - If the output is
` IIIT, Allahabad `

Then
 Vector 0 = ``
 Vector1 = `IIIT, Allahabad`
 Vector2 = ``

The rendering technique for the following html tags is presented in this paper.

HTMLTags are:
`<a> `, ` `, `<body> </body>`, `
`, `<center> </center>`, ` `, `<hr>`, `<h1> to <h6>`, ``, `<form> </form>`, `<input>`, `<select>`, `<option>`, `<textarea> </textarea>`, `<u> </u>`, ``, ``, ``, `<dir>`.

This web browser supports three types of images: png image, jpeg image, gif image.

In this paper, web page adaptation approach has the following seven steps:

- Lexical Analysis
- Parsing
- Block identification
- Block Generation
- New overview page Generation
- Block extraction
- Display of block

V. PROPOSED SOLUTION

In the previous work of server side content adaptation, contents are stored in the content cache in some format. There are some css programs made according to the specifications of the client devices. Whenever server gets the client's request, server detects the device, searches the requested content and the style sheet for the detected device. Then the server sends the customized content using the style sheet to the client.

In this paper it is proposed that no style sheet will be used for the customization of content i.e. the work of searching style sheets and customizing the content is reduced. Here already customized contents for the client devices are stored in a single body. Whenever the server gets the client's request, the server has to detect the device and only search the single program to get the content satisfying the client's specification. Now the single source containing the different forms of contents is an html program consisting of the predefined html tags. After the device detection three solutions are sought out for searching the appropriate content for the device and sending it to the device.

Outline of the proposed solution is depicted in the following Fig. 3.

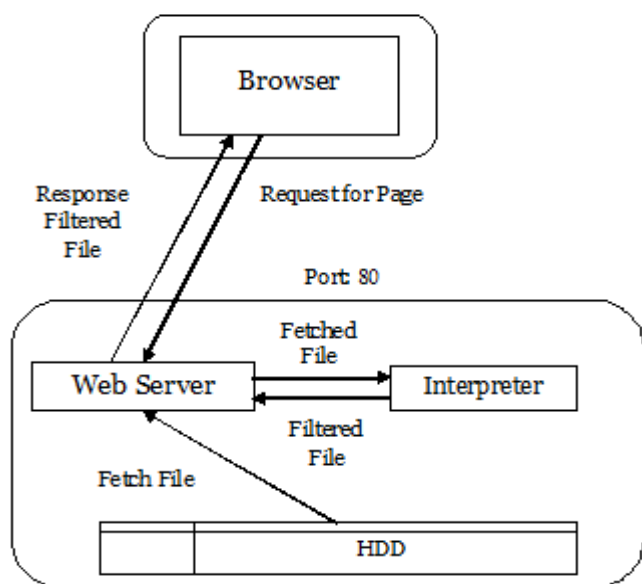


Fig. 3. Outline

A. Methodology

The implementation steps for the preliminary solution are given below:

Step-I:

Client requests for a web page to the server. The request is sent through an interface.

Whenever a web browser requests a page from a server, it sends an HTTP request to the server. The server in turn sends back an HTTP response. Two standard HTTP request methods are GET & POST.

The HTTP header contains the following:

From- Gives the name of the requesting user.

Accept- Contains a semicolon-separated list of representation schemes which will be accepted in the response to this request. Parameters on the content type are extremely useful for describing resolutions, colour depths, etc.

Accept-Encoding- Lists the Content-Encoding types which are acceptable in the response.

Accept-Language- Lists the Language values which are preferable in the response.

User-Agent- Gives the software program used by the original client.

Referer- This optional header field allows the client to specify, for the server's benefit, the address (URI) of the document (or element within the document) from which the URI in the request was obtained.

Authorization-This field contains authorization information.

Charge-To- contains account information for the costs of the application of the method requested

If-Modified-Since- Used with GET method to make it conditional: if the requested document has not changed since the time specified in this field the document will not be sent

Pragma- Pragma directives should be understood by servers to which they are relevant

Step-II:

Specifications rather limitations of client device are found from the information in the http header.

Clients are identified by the server through the User-Agent header. The User-Agent header contains information about the browser and the operating system making the request, and sometimes contains hardware information.

Identification of Client's Limitations

A device detection library or database installed on the web server (or a remote web service) to identify the device accessing the web site and return its capabilities. This set of capabilities allows the web developer to fine-tune the

resulting page to match the device's capabilities. The current leaders in device detection are WURFL and Device Atlas.

Step-III:

Now the web server fetches the file from hard disk.

The file contains all the contents customized for different devices within comment tags. Customized contents for each device are preceded by an identical markup according to the device name which helps in searching the content. e.g if the content is for mobile then it is preceded by `<mscript>`.

Step-IV:

Web server forwards the fetched file to the interpreter.

Step-V:

The interpreter filters the content according to device specification found in HTTP header. This process involves uncommenting the codes defining the content for the requesting client. The html page is read line by line and the markup (e.g., `<mscript>`) is searched for that is appropriate for the device. Then code marked up is uncommented. The new html page generated consists of the content customised for the client for adaptation and also the commented portions i.e., the customised contents for other devices.

Step-VI:

The interpreter forwards the filtered file to the server.

Step-VI:

The server returns the file to the client.

Step-VII:

The file is rendered in the client browser

The file generated from the above solution contains content for the client as well as the commented portion which contains content for the other devices. So the above solution is not the optimum solution since the file size is large and it consumes much memory space i.e., there lies a problem of client overload. To decrease the file size of the generated page another solution is found. In this solution, the file consists of only the required portions i.e., the commented portions for the other devices are removed.

Now in this solution also some ambiguities are found while fetching only the corresponding code for the client device. The condition that until the end of the comment tag is reached the code is fetched. Now there are so many end of comment tags. To identify the comment tag upto which the code is to be fetched, the code for the required content is also ended with an identical word followed by the end of comment tag, e.g., if the client is a mobile phone then the corresponding customised web content is bound within `<!-- mscript and /mscript-->`.

The working principle for the final solution is given below.

Repeat all the steps of the previous solution except the filtering process of the interpreter.

Filtering process

The file is read line by line and copied until a comment is found i.e., copy content from index 0 to the index of first `<!--`. This portion comprises of the common code for all devices.

Now to acquire the customized portion for the requesting device, the following rule is used. If the requesting client is mobile (identical markup for mobile is `mscript`) then index of `<!--mscript` is searched and length of `<!--mscript` is found. Now code from [index of (`<!--mscript`)+Length of (`<!--mscript`)] to index of (`/mscript-->`) is fetched.

Then again the last portion is also common to all devices. So the last position of end of comment tag is found. The code from last comment tag to the end is fetched.

Finally, after the server side adaptation the requested page is generated. The generated file is then sent to the client by the server through HTTP response.

B. Advantages of the proposed solution

- Predefined html tags are needed only.
- Searching time reduced.
- Less client-side work load
- Less download volume for the client
- To use this adaptation method, it is needed only to copy the single resource

C. Limitations / Problems found in this type of content adaptation

- It requires the web developer to use a device detection solution which needs to be kept up to date. Most device detection solutions are now commercial.
- Multi-source authoring is not a cost-efficient and feasible solution, especially for mobile services consisting of numerous pages.

VI. CONCLUSION

The advent of mobile web has made content adaptation an interested research topic. Mobile web holds a lot of promise, by providing increased mobility and encouraging device independence. Hence there is a need to use efficient adaptation techniques to adapt the existing websites to the small screen of a mobile device. Various adaptation techniques are there. In this paper, a generic method for server side content adaptation is proposed where the different formats of the content are stored as a single program. The proposed solution not only reduces work overload but also helps the mobile devices to render the contents easily.

REFERENCES

- [1] Bellinzona D, Vitali P "Multimedia content adaptation through tag libraries". Database and Expert Systems Application, 2008. DEXA '08. 19th International Conference on, DEXA '08. 19th International Conference on Database and Expert Systems Application, 2008, pp 716-720

- [2] Niranjan Mangal “Web content customisation and adaptation in mobile devices”. Indian Institute of Information Technology, Allahabad, 2007
- [3] Mehregan Mahdavi, Hamid Khordadi and Mohammad-Hassan Khoobkar “Web Transcoding for Mobile Devices Using a Tag-Based Technique”. Department of Computer Science and Engineering, University of Guilan, Iran 1, Department of Information Technology, University of Shiraz, Iran 2, Faculty of Computer Science and Engineering, Islamic Azad University, Lahijan Branch, Iran, 2010.
- [4] Fu Xiao-ling, Xiong Xiao-bo, Wang Dian-lai, Liu Feng, Implementation of a mobile learning platform based on WAP, International Conference on Computer Science and Software Engineering, 2008.
- [5] Stephen J.H. yangl, Irene Y.L. chen, Rick chen, “Applying Content Adaptation to Mobile Learning”, IEEE, 2007.
- [6] Hamed Ahmadi, Jun Kong, “Efficient Web Browsing on Small Screens”, ACM , 2008.
- [7] World Wide Web Consortium (W3C) www.w3.org/
- [8] Xing Xie, Yu Chen and Hong – Jiang Zhang, “Adapting Web pages for small screen devices”, IEEE Internet Computing, , January 2005, Vol. 9, pp. 50-56.
- [9] Alembik Web Site <http://alembik.sourceforge.net>