ADAPTING HTML CONTENTS FOR WAP DEVICES USING JAVA SERVLETS

A DISSERTATION

Submitted in partial fulfilment of the
requirements for the award of the degree
of
MASTER OF TECHNOLOGY
in
COMPUTER SCIENCE AND TECHNOLOGY

By

MANISH SHARMA

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
UNIVERSITY OF ROORKEE
ROORKEE–247 667 (INDIA)

FEBRUARY, 2001

CANDIDATE'S DECLARATION

I hereby certify that the work, which is being presented in the dissertation entitled **"ADAPTING HTML CONTENTS FOR WAP DEVICES USING JAVA SERVLETS"**, in partial fulfillment of requirements for the award of the degree of Master of Technology, in Computer Science and Technology, submitted in the Department of Electronics and Computer Engineering, University of Roorkee, Roorkee, is an authentic record of my own work carried out from August 2000 to February 2001, under supervision of **Dr. Padam Kumar**, Professor, Department of Electronics and Computer Engineering, University of Roorkee, Roorkee.

The matter embodied in this dissertation has not been submitted by me for the award of any other degree.

Date : 26/2/2001

Place: Roorkee

(MANISH SHARMA)

CERTIFICATE

This is certified that the above statement made by the candidate is true to the best of my knowledge.

Date: 26/2

Place : Roorkee

(Dr. Padam Kumar)
Professor,
Deptt. of Electronics & Computer Engg.
University of Roorkee
Roorkee (India) –247667.

ACKNOWLEDGEMENTS

Firstly I wish to express my deep sense of indebtedness and sincerest gratitude to my guide, **Dr. Padam Kumar**, for extending his ever encouraging and affable guidance and encouragement throughout the progress of this dissertation work. He has displayed unique tolerance and understanding at every step of progress and encouraged me incessantly. I deem it my privilege to have carried out my dissertation work under his able guidance.

I wish to acknowledge my parents and friends who had spoken the words of love and encouragement. At last I gratefully acknowledge my deep indebtedness to all other persons who directly or indirectly helped me during the whole period of my stay at this university.

(MANISH SHARMA)

ABSTRACT

After the world wide web, the next big challenge to the Internet is mobile access. More and more information is available on the Internet and intranets and mobile users will also need access to it. These devices are characterized by low network bandwidth, high latency time, small display size, low computational capability. Wireless Application Protocol (WAP) together with Wireless Mark-up Language (WML) constitute an open architecture for mobile web services designed with these limitations in mind. To date, most web resources are authored in HTML with the desktop as the client device where resource is hardly a constraint. For the WAP devices to access these web documents, contents need to be converted into WML keeping in mind the resource constraints of the mobile devices. In this thesis we have described a content adaptation technique applied to HTML documents from a specific domain area - news web sites, since receiving news updates is one of the most important application of mobile Internet access. We have implemented a content adaptation system that extracts the most important information from the news sites, converts them into WML for display on WAP devices. As test platforms to display WML content, we have used WAP device simulators. This approach gives the mobile users transparent access to their familiar web pages from their mobile phones and other mobile devices.

CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Motivation

The rapid growth of Web services has led to a situation where companies and individuals rely more and more on material that is available on the Internet and intranets. An increasing number of people use Web services both at work and at home. The next step is to gain access to Web services for mobile users too. Already before WAP some simple interactive services have been available on mobile phones. These services are based on the GSM short message service (SMS) and include schedules, news, sport results, weather forecasts and so on. Although the available SMS-based services are quite awkward to use, they have become very popular. This indicates a need for additional mobile Web services. Access itself will probably be the killer application for the mobile Internet[1].

Wireless Application Protocol (WAP) together with Wireless Markup Language (WML) constitute an open architecture for mobile Web services. They make it possible to provide Wireless markup-language based services for different mobile devices equipped with WML browsers. The selection of WAP devices is expected to range from mobile phones to palmtop computers. The international specification work on WAP is still going on (February 2000) in the WAP Forum and several details must still be worked out[2]. The first WAP compliant devices have already been released on the market and more are to come.

How should the service providers create services to the growing variety of mobile clients? There are two different approaches to bringing Internet services to WAP phones and other mobile devices. First approach, known as, the mobile-aware approach, is to design and implement totally new services that are specially designed for mobile users. In the future there will be a need for Web services that are specially targeted for mobile users. But it would be economically more beneficial if the already existing Internet information were made accessible with WAP devices. The second and a more generic approach, known as mobile-transparent approach, is to develop techniques with which current Internet services can be converted transparently and in real time suitable for mobile users.

But to date, most web resources are authored in HTML with the desktop as the client device. These devices have large memory, large network bandwidth and large computational power. For the WAP devices to access these web documents, contents need to be converted into WML. But all the contents in the HTML document converted into WML may not be rendered properly because of the physical limitations like low network bandwidth, high latency time, small display size, low computational capability of the WAP devices. Therefore one needs to adapt the content of the HTML page on the fly keeping in mind the resource constraints of the client device. If it is possible to convert services transparently to mobile users, the service providers will save a lot in implementation and maintenance costs.

1.2 Our work

We have implemented a system based on the mobile- transparent approach. Our mobile-transparent solution is applied to a specific domain – news sites. Receiving news updates is one of the most important application of internet access through mobile phones. Our solution is an HTML-WML conversion proxy server. Since, newspaper sites provide new contents within a specific HTML structure, domain dependent rules for content adaptation at such sites are feasible to be maintained. The rules can be integrated to the site itself. The users access the newspaper sites through proxy servers where the content adaptation is done. The proxy server, on receipt of a request from a WAP client fetches the HTML page from the original site (obtained from the requested URL), applies the relevant rules for adaptation. Relevant information is extracted from the HTML page using the extraction rules of adaptation, converted to WML and sent back to the WAP client. This approach gives the mobile users transparent access to their familiar web pages from their mobile phones and other mobile devices.

1.3 Organization of the thesis

Chapter 2 gives an introduction to WAP architecture and the Wireless Application Environment (WAE) that enable mobile users to access to internet services. We also introduce the concept of content adaptation and a brief outline of various content adaptation techniques that are being employed currently. Chapter 3 focuses on the content adaptation technique that we have employed to our specific domain area. It describes the basic framework of our content adaptation system, its key components and the extraction rules we have used to extract relevant information from the HTML document.

In chapter 4, we present the implementation details of our system, the content adaptation methodology and some of the results we have obtained. Finally, in chapter 5 we conclude with a summary and directions for future work.

CHAPTER 2

WIRELESS APPLICATION PROTOCOL

In this chapter we describe the WAP architecture, the Wireless Application Environment (WAE) model. We also describe the various Content Adaptation techniques that have been employed.

The Wireless Application Protocol is a de-facto industry standard for internet access to wireless devices. It was developed by the WAP Forum[3], a group founded by Nokia, Ericsson, Phone.com (formerly Unwired Planet), and Motorola. According to the WAP Forum, the goals of WAP are to be:

- Independent of wireless network standard.
- Open to all.
- Proposed to the appropriate standards bodies.
- Scalable across transport options.
- Scalable across device types.
- Extensible over time to new networks and transports.

WAP defines a communications protocol as well as an application environment. In essence, it is a standardized technology for cross-platform, distributed computing. WAP is very similar to the combination of HTML and HTTP except that it adds in one very important feature: optimization for low-bandwidth, low-memory, and low-display capability environments. These types of environments include PDAs (PERSONAL

IGITAL ASSISTANT) , wireless cellular phones, pagers, and virtually any other ommunications device.

.1 WAP and the Web

rom a certain viewpoint, the WAP approach to content distribution and the Web pproach are virtually identical in concept. Both concentrate on distributing content to emote devices using inexpensive, standardized client software. Both rely on back-end ervers to handle user authentication, database queries, and intensive processing. Both se markup languages derived from SGML (STANDARD GENERALIZED MARKUP ANGUAGE) for delivering content to the client. In fact, as WAP continues to grow in upport and popularity, it is highly likely that WAP application developers will make se of their existing Web infrastructure (in the form of application servers) for data torage and retrieval.

WAP allows a further extension of this concept as existing 'server' layers can be eused and extended to reach out to the vast array of wireless devices in business and ersonal use today.

WAP uses some new technologies, however the overall concepts are similar to amiliar access mechanism of a URL from a desktop browser. WAP client applications ake requests very similar in concept to the URL concept in use on the Web. As a eneral example, consider the following explanation (exact details may vary on a vendor-o-vendor basis).

A WAP request is routed through a WAP gateway which acts as an intermediary etween the 'bearer' used by the client (GSM, CDMA, TDMA, etc.) and the computing network that the WAP gateway resides on (TCP/IP in most cases). The gateway then processes the request, retrieves contents or calls CGI scripts, Java servelets, or some other dynamic mechanism, then formats data for return to the client. This data is formatted as WML (Wireless Markup Language), a markup language based directly on XML[4] (EXTENSIBLE MARKUP LANGUAGE). Once the WML has been prepared (known as a deck), the gateway then sends the completed request back (in binary form due to bandwidth restrictions) to the client for display and/or processing. The client retrieves the first card off of the deck and displays it on the monitor.

The deck of cards metaphor is designed specifically to take advantage of small display areas on handheld devices. Instead of continually requesting and retrieving cards (the WAP equivalent of HTML pages), each client request results in the retrieval of a deck of one or more cards. The client device can employ logic via embedded WMLScript (the WAP equivalent of client-side JavaScript) for intelligently processing these cards and the resultant user inputs. To sum up, the client makes a request. This request is received by a WAP gateway that then processes the request and formulates a reply using WML. When ready, the WML is sent back to the client for display. As mentioned earlier, this is very similar in concept to the standard stateless HTTP transaction involving client Web browsers.

2.1.1 Communications Between Client and Server

The WAP Protocol Stack[3] is implemented via a layered approach (similar to the OSI network model). These layers consist (from top to bottom) of:

1. Wireless Application Environment (WAE)
2. Wireless Session Protocol (WSP)
3. Wireless Transaction Protocol (WTP)
4. Wireless Transport Layer Security (WTLS)
5. Wireless Datagram Protocol (WDP)
6. Bearers (GSM, IS-136, CDMA, GPRS, CDPD, etc.)

According to the WAP specification, WSP offers means to:

- provide HTTP/1.1 functionality:
- extensible request-reply methods,
- composite objects,
- content type negotiation
- exchange client and server session headers
- interrupt transactions in process
- push content from server to client in an unsynchronized manner
- negotiate support for multiple, simultaneous asynchronous transactions

WTP provides the protocol that allows for interactive browsing (request/response) applications. It supports three transaction classes: unreliable with no result message, reliable with no result message, and reliable with one reliable result message. Essentially, WTP defines the transaction environment in which clients and servers will interact and exchange data. The WDP layer operates above the bearer layer used by your communications provider. Therefore, this additional layer allows applications to operate transparently over varying bearer services. While WDP uses IP as the routing protocol, unlike the Web, it does not use TCP. Instead, it uses UDP (User Datagram Protocol) which does not require messages to be split into multiple packets and sent out only to be reassembled on the client. Due to the nature of wireless communications, the mobile application must be talking directly to a WAP gateway (as opposed to being routed through myriad WAP access points across the wireless Web) which greatly reduces the overhead required by TCP.

For secure communications, WTLS is available to provide security. It is based on SSL and TLS.

2.2 The Wireless Application Environment (WAE)

The Wireless Application Environment[2] (WAE) is a general-purpose application environment based on a combination of World Wide Web (WWW) and Mobile Telephony technologies. The primary objective of the WAE effort is to establish an interoperable environment that will allow operators and service providers to build applications and services that can reach a wide variety of different wireless platforms in an efficient and useful manner. WAE includes a micro-browser environment containing the following functionality:

Wireless Markup Language (WML) - a lightweight markup language, similar to HTML, but optimized for use in hand-held mobile terminals, such as mobile telephone and PDAs;

WMLScript - a lightweight scripting language, similar to JavaScript;

Wireless Telephony Application (WTA, WTAI) - telephony services and interfaces; and Content Formats - a set of well-defined data formats, including images, phone book records and calendar information.

WAE adopts a model that closely follows the WWW model. All content is specified in formats that are similar to the standard Internet formats. Content is transported using standard protocols in the WWW domain and an optimized HTTP-like protocol in the wireless domain. WAE has borrowed from WWW standards including authoring and publishing methods wherever possible. The WAE architecture allows all content and services to be hosted on standard Web origin servers that can be incorporate proven technologies (e.g., CGI). All content is located using WWW standard URLs.

WAE enhances some of the WWW standards in ways that reflect the device and network characteristics. WAE extensions are added to support Mobile Network Services such as Call Control and Messaging. Careful attention is paid to the memory and

CPU processing constraints that are found in mobile terminals. Support for low bandwidth and high latency networks is included in the architecture as well.

WAE assumes the existence of gateway functionality responsible for encoding and decoding data transferred from and to the mobile client. The purpose of encoding content delivered to the client is to minimize the size of data sent to the client over-the-air as well as to minimize the computational energy required by the client to process that data. The gateway functionality can be added to origin servers or placed in dedicated gateways as illustrated in Figure 2.1

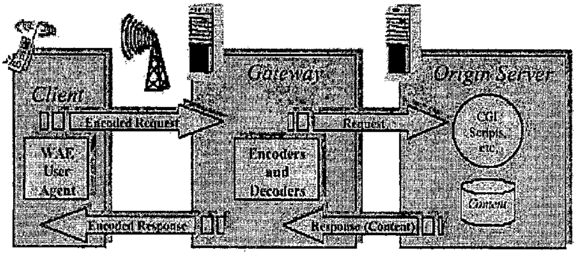


Figure 2.1 WAE Logical Model (Source: www.wapforum.org)

.3 WML

WML[5] is a tag-based document language similar to HTML. It is specified s an XML document type. It was designed with the following constraints in mind:

small display.

limited user-input facilities

narrow band connections

limited memory resources

limited computational resources.

WML implements a card and deck metaphor. An interaction with the user is described ı a set of cards, which are grouped together into a document referred to as a deck. ,ogically a user navigates through a set of cards. A card the basic unit of display. To ıake it readable, it is advisable to break up a document into cards with less content so ıat user comprehendibility is more. Deck is basic unit of transfer when a request for a ocument is made. This is limited by the resource capability of the WAP device.

Some of the features of WML are:

Support for text and images.

WML provides the authors with means to specify text and images to the user. 'his may include layout and presentation hints. It provides a set of mark-up elements ıcluding various emphasis elements (e.g. bold, italic, big, etc.); tab columns that support imple tabbing alignment. Figure 2.2 shows a simple WML card as it would appear on a ıobile phone equipped with a WML browser. The WML source is also shown.

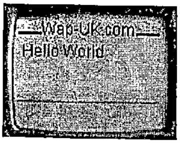


Figure 2.2

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
 <card id="Card1" title="Wap-UK.com">
    <p>
    <!-- Hello World example -->
    Hello World
   </p>
 </card>
</wml>
```

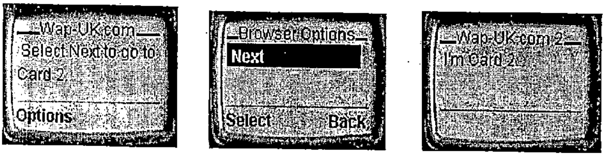
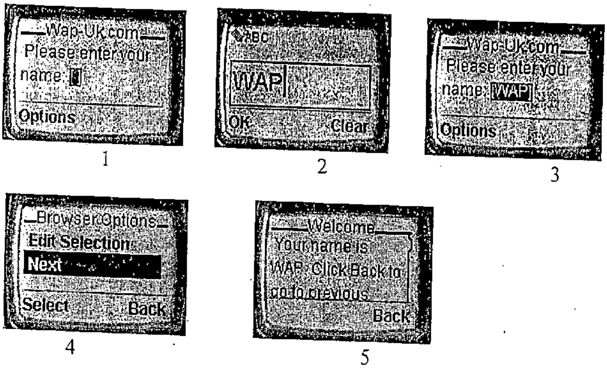


Figure 2.3

Inter-card navigation

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card id="Card1" title="Wap-UK.com">
   <do type="accept" label="Next">
     <go href="#Card2"/>
   </do>
   <p>
     Select Next to go to Card 2.
   </p>
  </card>
  <card id="Card2" title="Wap-UK.com">
   <p>
     I'm Card 2.
```

```
    </card>
  </wml>
```



1

Wap-Uk.com
Please enter your name:
Options

2

ABC
WAP
OK Clear

3

Wap-Uk.com
Please enter your name: WAP
Options

4

Browser Options
Edit Selection
Next
Select Back

5

Welcome
Your name is WAP. Click Back to go to previous
Back

Figure 2.3

Input from user

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
  <card id="main" title="Wap-Uk.com">
   <do type="accept" label="Next">
     <go href="#wel"/>
   </do>
   <p>
    Please enter your name:
    <input type="text" name="name"/>
   </p>
 </card>
  <card id="wel" title="Welcome">
   <do type="prev" label="Back">
    <prev/>
   </do>
   <p>
    Your name is $(name). Click Back to go to previous page.
   </p>
  </card>
</wml>
```

- Navigation and History Stack

Navigation mechanisms include HTML-style hyperlinks, inter-card navigation elements, as well as history navigation elements. Figure 2.3 shows a simple example of navigation between two cards belonging to the same deck.

- Support for user input

It supports several elements to solicit user input. It also includes a set of input controls. For example, WML includes a text entry control that supports text and password entry. WML includes an option selection control that allows the author to present the user with a list of options that can set data, navigate among cards, or invoke scripts. Figure 2.4 shows an example of user of capturing user input. The sequence of events are numbered from 1 to 5.

2.4 WMLSCRIPT

The purpose of WMLScript[6] is to provide client-side procedural logic. It is based on ECMAScript (which is based on Netscape's JavaScript language), however it has been modified in places to support low bandwidth communications and thin clients. The inclusion of a scripting language into the base standard was an absolute must. While many Web developers regularly choose not to use client-side JavaScript due to browser incompatibilities (or clients running older browsers), this logic must still be replaced by additional server-side scripts. This involves extra roundtrips between clients and servers which is something all wireless developers want to avoid. WMLScript allows code to be built into files transferred to mobile client so that many of these round-trips can be eliminated. According to the WMLScript specification, some capabilities supported by WMLScript that are not supported by WML are:

- Check the validity of user input

Access to facilities of the device. For example, on a phone, allow the programmer to make phone calls, send messages, add phone numbers to the address book, access the SIM card etc.

- Generate messages and dialogs locally thus reducing the need for expensive round-trip to show alerts, error messages, confirmations etc.
- Allow extensions to the device software and configuring a device after it has been deployed.

WMLScript is a case-sensitive language that supports standard variable declarations, functions, and other common constructs such as if-then statements, and for/while loops. Among the standard's more interesting features are the ability to use external compilation units (via the use url pragma), access control (via the access pragma), and a set of standard libraries defined by the specification (including the Lang, Float, String, URL, WMLBrowser, and Dialogs libraries). The WMLScript standard also defines a bytecode Interpreter since WMLScript code is actually compiled into binary form (by the WAP gateway) before being sent to the client.

2.5 CONTENT ADAPTATION

Content Adaptation is defined as the process to adapt the original content to suit the network bandwidth or terminal characteristics of the access device. This may include data transcoding i.e., changing the format into another format that is suitable for display on the access device. For example, some client devices may not be able to display colour GIF images because of lack of viewing or rendering software or the constraint of hardware display capability, such as black-and-white screen. In such cases, there is a need to transcode the original image into an appropriate format like colour-to-grayscale transformation, so that they can be viewed on the client device. In case of WAP architecture discussed above, the HTML content need to be converted to WML for display on WAP device.

Content Adaptation can be performed at the origin sever, or at an intermediate proxy between the client and the server or at the client side itself. All the three approaches have been employed for content adaptation. We examine these three cases by studying the existing content adaptation systems.

Client-based adaptation.

Some client devices adapt content the device. For example, Windows CE devices change colour depth (for example, from 24-bit colour to 4-bit grey-level.) of images. The drawbacks are that these devices have low network bandwidth, restricted computational power which makes content adaptation at the device very slow, or even impossible.

Server-based adaptation

In a server-based adaptation architecture, the server is responsible for discovering what the client capabilities are and how much bandwidth is available. It then decides on the best adaptation strategy. Using the server-based architecture has the advantage that it allows both static (off-line) and dynamic (on-the-fly) content adaptation. The former refers to an authoring post-processing situation, where the adaptation automatically creates multiple versions on the authored content, at anytime after the content has been created; the latter refers to performing an on-the-fly adaptation as each request comes in. The server architecture provides more author control since the adaptation can be tied to the content authoring process, allowing the author to provide hints on the adaptations for different circumstances. An author can also preview the adapted outcome under different viewer preferences and conditions. This greatly eases the authoring process for heterogeneous environments, as the author need only create the content once, and the adaptive content delivery system automatically ensures the appropriate content version is delivered. For a secured environment, such as in e-commerce applications, pages are usually encrypted, in which case only the server can perform adaptation.

Placing the adaptation on the server also has drawbacks. It is generally desirable for a client to download from a server in a nearby geographical location, since the shortened distance and fewer hops make the download time faster. In the server-based architecture, to achieve this would entail having geographically distributed servers, thus server design needs to consider data synchronization. Moreover, the adaptation results in additional computational load and resource consumption on the server; thus the server design needs to take into account load balancing as well. The static approach generates multiple versions of the content, thus making content management more cumbersome and requiring more storage.

A server-based content adaptation system has been proposed by Rakesh Mohan et.al.[7] that is an extension of a Web server. They propose a content adaptation framework that dynamically accounts for resource requirements of the web page and its individual components. It selects from a number of different possible transcoded versions of content, ones that provide the "best value" within the constraints of a client's resource. This system makes use two key technologies:

1) A progressive data representation scheme called the Infopyramid[8]. Content items on a web page are transcoded into multiple resolution and modality versions so that they can be rendered on different devices. For example, a video item is transcoded into a set of images so that it can be rendered on a device not capable of displaying video.

2) A customizer that selects the best versions of content items from the Infopyramids To meet the client resources while delivering the "most value". The customizer allocates resources on the client among the items in the document. This resource allocation results in the selection of the appropriate resolution or modality of the content items. If the client has limited resources (such as PDA or cell-phone), some of the content items may not get any resources assigned and thus will not be delivered to the client.

The content is authored in XML, allowing the author to provide more information to the transcoding customizing systems than can be deduced from an HTML page.

Proxy-based adaptation

In a proxy-based adaptation, the client connects through a proxy which then makes the request to the server on behalf of the client. The proxy intercepts the reply from the server, decides on and performs the adaptation, and then sends the transformed content back to the client. It is usually assumed that the bandwidth between the proxy and the server is much higher than that between the client and the proxy, so that the time to download the original content from the server to the proxy is negligible. This is true, for example, when the proxy resides at the point-of-presence (POP) of an Internet Service Providers, and when the client is connecting through a slow modem or wireless modem.

A proxy-based architecture makes it easy to place adaptation geographically close to the clients. Adapting on the proxy means that there is no need to change existing clients and servers, and it achieves economy of scale more than a server architecture since each proxy can transform content for many servers. The proxy can transform existing Web content so that existing content does not have to be re-authored.

Because a proxy takes the content from many servers, there are many documents with widely varying appearances, created with many different authoring tools. Hence, in proxy architecture, there is less author control on the outcome of the adaptation, and it is difficult to determine what alteration "looks good" for any general content. Thus the author of the Web page may find the resulting transformation objectionable, and it may be difficult to reliably ensure that a transformed page will look aesthetically pleasing. For secured or proprietarily encoded content such as RealNetworks [9] streaming media, the organization deploying the proxy will need to coordinate with the service or content provider in order to access the content for performing adaptation.

The issue of copyright infringement becomes significant in a proxy-based system, since an author has little control of the content adaptation. For example, a common URL filter is that of blocking advertisement logos. Considering that many free Web services rely on advertising revenues, these content providers and their advertising partners will likely be displeased by the decreased hit rates of the ads due to the ad-blocking filter.

Several commercial proxies have been deployed for the purpose of adapting Web content for universal access [10][11]. Because of these difficult issues, the commercial adaptation proxies have either targeted specific clients and usage environments or have partnered with portal and content provider companies. Partnering with content providers also circumvents the copyright issue.

We look at some proxy-based systems. First one is known as Digestor proposed by Bickmore and Schilit[12]. Digestor is a software system which automatically re-authors arbitrary documents from the world-wide web to display appropriately on small screen devices such as PDAs and cellular phones, providing device-independent access to the web. Digestor is implemented as an HTTP proxy which dynamically re-authors requested web pages using a heuristic planning algorithm and a set of structural page transformations to achieve the best looking document for a given display size. It uses various re-authoring techniques like text-summarization, providing header or titles to block of text, image-reduction etc. The adaptation technique relies on the user providing the device characteristics like display size, color-depth, default browser font sizes etc. Once a user has configured the system, they can start retrieving documents from the web.

Eija Kaasinen et.al.[13] have developed an HTML-to-WML conversion proxy, which converts HTML-based Web contents automatically and on-line to WML. They do not employ any content adaptation techniques to the original HTML page. Their studies Have indicated that if HTML-based web services follow certain guidelines, they can be converted automatically to WML and adapted to client devices. In principle these guidelines already exist as W3C Web Content Accessibility Guidelines and W3C note for HTML 4.0 Guidelines for mobile access.

Wei-Ying Ma et.al.[14] have proposed a framework to provide a broad range of web content adaptation for all different types of devices under heterogeneous and changing network conditions. This system, known as Adaptive Content Delivery System, uses the following key technologies:

Media processing and analysis algorithms to support content adaptation.

Policies for determining when and how to launch a particular content adaptation algorithm based upon various conditions.

A mechanism for reliably detecting the software and hardware capabilities of a client device.

A standard approach for defining user preferences and a mechanism for tracking them from session to session.

A way to effectively measure the characteristics of the current network connection between a client and a server.

The basic framework for their adaptive content delivery system of three main modules - user/client/network-discovery, Decision Engine, and content adaptation algorithm modules along with interfaces to administration and authoring tools and system devices such as client, proxy, and server.

The user/client/network-discovery module. It detects and collects all necessary information that the Decision Engine needs to know in order to dispatch a particular content adaptation algorithm.

The Decision Engine. The inputs to the Decision Engine are a set of media objects contained in a document and the information about their content types, content lengths, and purposes of usage in a Web page. The Decision Engine first checks the user preferences to see if it needs to remove or substitute any redundant objects in order to save bandwidth. Other information for making decisions, such as how much a user is willing to trade off image quality for download time, is acquired by the Decision Engine at this time. The Decision Engine further checks the client capability and characteristics of the network that the client uses to connect to the server. Based upon the collected information, the Decision Engine then determines if it needs to launch a particular content adaptation algorithm for a particular object.

Content Adaptation Algorithms. They are a set of media processing and analysis algorithms to support content adaptation.

In this chapter we have given a short overview of the basic WAP technology. An interesting area of research is integration of already existing WEB structure to WAP applications. Our work is centered around developing such an application which will enable a WAP user to access Internet content from a particular site. Our main emphasis has been to develop domain dependant methodologies to provide better service to a WAP user.

CHAPTER 3

CONTENTS ADAPTATION FOR WAP APPLICATION

Internet access to wireless mobile devices involves two basic issues:

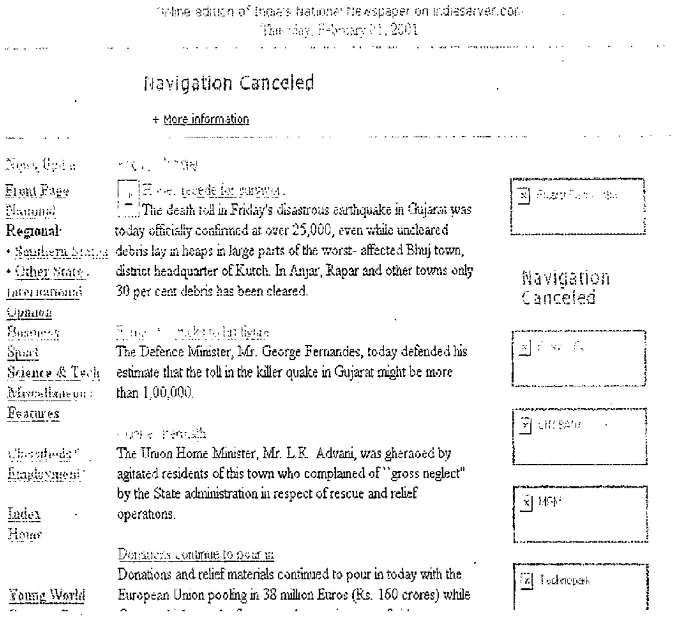
1. The underlying network technology.
2. Capability of client devices.

There are physical limitations like small display size, resolution, memory and low computational capability. WAP provides a set of communication protocols and application environment that makes it possible for to provide web resources that are authored in WML to WAP-enabled devices. In the future there will be a need for web services that are specifically targeted for mobile users. But it would be beneficial if existing web resources (HTML based) are made accessible to WAP enabled devices. By applying content adaptation techniques, we aim to provide access to existing HTML based contents to WAP devices in a transparent manner. In the following sections we describe the basic framework of our content adaptation system, the extraction rules which we apply on a HTML page to extract relevant information and the transcoding system which converts the HTML to WML.

3.1 OUR WORK

Content adaptation can be performed at the content's origin server or at an intermediate proxy server between the client and origin server. In the WAP WAE architecture described in the previous chapter (ref. Figure 2.1) such a proxy exists whose primary function is to encode content into binary format for over the air transmission to the mobile client. Our approach is based on the latter architecture where the content from the origin server (HTML) is adapted to WML by our Content Adaptation System and encoded into binary format by the proxy-server before transmitting it to the client.

As discussed in the previous chapter, it has been observed content adaptation techniques have been applied to web-contents in general, without any specific domain knowledge, i.e., without assuming any particular underlying structure of the web document. However web documents from a specific domain sometimes have a common underlying structure though they may be rendered differently. If we choose a specific domain area, for which we may be able to describe an underlying structure of the documents that is common to them, then content adaptation technique can be greatly simplified and implemented more efficiently for those web documents. This is significant since WAP devices are more customer centric in nature and the access patterns are more focussed. Such a focussed domain is news access. We have chosen to apply content adaptation techniques with news documents as our domain area, i.e., web-sites which provide news (E.g. www.the-hindu.com, www.timesofindia.com etc.) since one the important applications of mobile access to internet is getting news updates. Figures 3.1, 3.2 show snapshots of two web sites, which provide news, as they appear on a standard HTML browser. Also shown in figure 3.3 is part of the HTML source code.



[illegible] needs for survival

The death toll in Friday's disastrous earthquake in Gujarat was today officially confirmed at over 25,000, even while uncleared debris lay in heaps in large parts of the worst- affected Bhuj town, district headquarter of Kutch. In Anjar, Rapar and other towns only 30 per cent debris has been cleared.

[illegible]

The Defence Minister, Mr. George Fernandes, today defended his estimate that the toll in the killer quake in Gujarat might be more than 1,00,000.

[illegible]

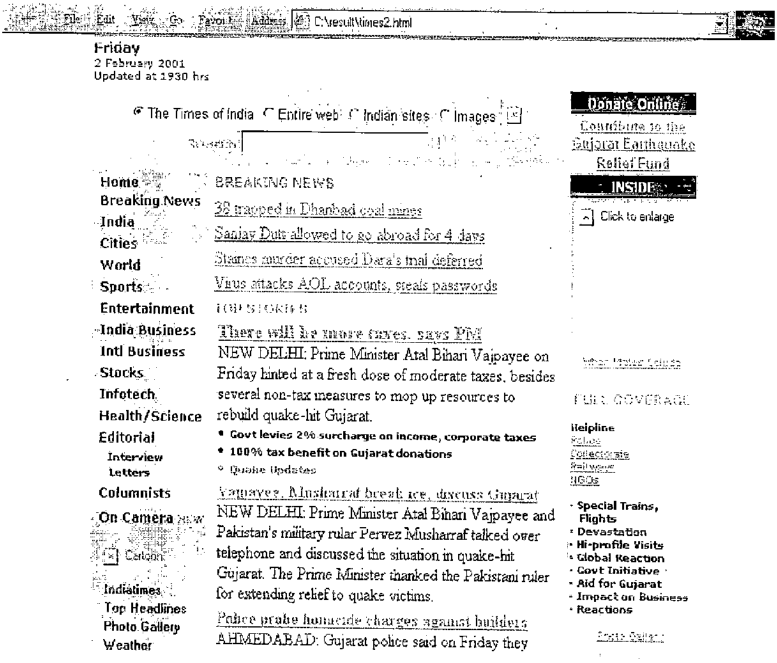
The Union Home Minister, Mr. L.K. Advani, was gheraoed by agitated residents of this town who complained of ``gross neglect" by the State administration in respect of rescue and relief operations.

Donations continue to pour in

Donations and relief materials continued to pour in today with the European Union pooling in 38 million Euros (Rs. 160 crores) while

Figure. 3.1 Snapshot of Hindu main page.

(source www.the-hindu.com)



Friday
2 February 2001
Updated at 1930 hrs

● The Times of India ○ Entire web ○ Indian sites ○ Images

Search

BREAKING NEWS

38 trapped in Dhanbad coal mines

Sanjay Dutt allowed to go abroad for 4 days

Staines murder accused Dara's trial deferred

Virus attacks AOL accounts, steals passwords

TOP STORIES

There will be more taxes, says PM

NEW DELHI: Prime Minister Atal Bihari Vajpayee on Friday hinted at a fresh dose of moderate taxes, besides several non-tax measures to mop up resources to rebuild quake-hit Gujarat.

- **Govt levies 2% surcharge on income, corporate taxes**
- **100% tax benefit on Gujarat donations**
- Quake Updates

Vajpayee, Musharraf break ice, discuss Gujarat

NEW DELHI: Prime Minister Atal Bihari Vajpayee and Pakistan's military rular Pervez Musharraf talked over telephone and discussed the situation in quake-hit Gujarat. The Prime Minister thanked the Pakistani ruler for extending relief to quake victims.

Police probe homicide charges against builders

AHMEDABAD: Gujarat police said on Friday they

Donate Online

Contribute to the Gujarat Earthquake Relief Fund

INSIDE

Click to enlarge

When Mother Earth...

FULL COVERAGE

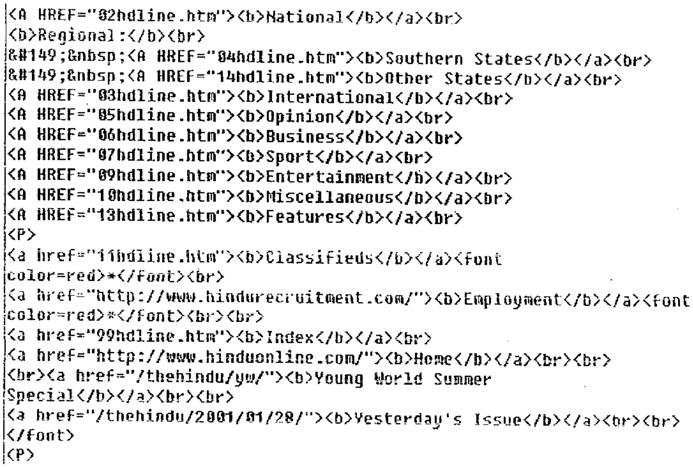
Helpline
Police
Collectorate
Railways
NGOs

- **Special Trains, Flights**
- **Devastation**
- **Hi-profile Visits**
- **Global Reaction**
- **Govt Initiative**
- **Aid for Gujarat**
- **Impact on Business**
- **Reactions**

Photo Gallery

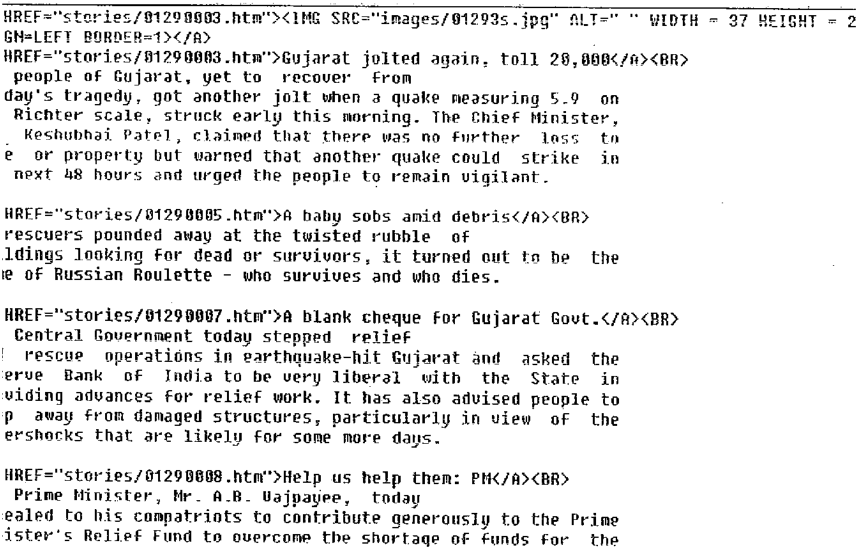
Figure 3.2 Snapshot of times of India web page

(source www.timesofindia.com)



```
<A HREF="02hdline.htm"><b>National</b></a><br>
<b>Regional:</b><br>
&#149;&nbsp;<A HREF="04hdline.htm"><b>Southern States</b></a><br>
&#149;&nbsp;<A HREF="14hdline.htm"><b>Other States</b></a><br>
<A HREF="03hdline.htm"><b>International</b></a><br>
<A HREF="05hdline.htm"><b>Opinion</b></a><br>
<A HREF="06hdline.htm"><b>Business</b></a><br>
<A HREF="07hdline.htm"><b>Sport</b></a><br>
<A HREF="09hdline.htm"><b>Entertainment</b></a><br>
<A HREF="10hdline.htm"><b>Miscellaneous</b></a><br>
<A HREF="13hdline.htm"><b>Features</b></a><br>
<P>
<a href="11hdline.htm"><b>Classifieds</b></a><font
color=red>*</font><br>
<a href="http://www.hindurecruitment.com/"><b>Employment</b></a><font
color=red>*</font><br><br>
<a href="99hdline.htm"><b>Index</b></a><br>
<a href="http://www.hinduonline.com/"><b>Home</b></a><br><br>
<br><a href="/thehindu/yw/"><b>Young World Summer
Special</b></a><br><br>
<a href="/thehindu/2001/01/28/"><b>Yesterday's Issue</b></a><br><br>
</font>
<P>
```

Figure. 3.3 HTML source of the web page shown in fig. 3.1



```
HREF="stories/01290003.htm"><IMG SRC="images/01293s.jpg" ALT=" " WIDTH = 37 HEIGHT = 2
GN=LEFT BORDER=1></A>
HREF="stories/01290003.htm">Gujarat jolted again, toll 20,000</A><BR>
 people of Gujarat, yet to  recover  from
day's tragedy, got another jolt when a quake measuring 5.9  on
 Richter scale, struck early this morning. The Chief Minister,
  Keshubhai Patel, claimed that there was no further  loss  to
e  or property but warned that another quake could  strike  in
 next 48 hours and urged the people to remain vigilant.

HREF="stories/01290005.htm">A baby sobs amid debris</A><BR>
rescuers pounded away at the twisted rubble  of
ldings looking for dead or survivors, it turned out to be  the
e of Russian Roulette - who survives and who dies.

HREF="stories/01290007.htm">A blank cheque for Gujarat Govt.</A><BR>
 Central Government today stepped  relief
  rescue  operations in earthquake-hit Gujarat and  asked  the
erve  Bank  of  India to be very liberal  with  the  State  in
viding advances for relief work. It has also advised people to
p  away from damaged structures, particularly in view  of  the
ershocks that are likely for some more days.

HREF="stories/01290008.htm">Help us help them: PM</A><BR>
 Prime Minister, Mr. A.B. Vajpayee,  today
ealed to his compatriots to contribute generously to the Prime
ister's Relief Fund to overcome the shortage of funds for  the
```

Figure. 3.4 Shows part of HTML source of web page shown in fig. 3.1

From figures 3.1 and 3.2 we observe that a typical page contains, on the left-hand side a set of links to various subsections of the news (which is evident from the HTML source shown in figure 3.3). Towards the center we have headlines followed by brief report of the headline. These headlines are used as links to page containing a more detailed report as can be seen from the HTML source shown in figure 3.4.

On analyzing it is found that though there are a number of elements in the HTML source code, for a newspaper site, a lot of it is redundant and of no use to mobile phone user. For example, a WAP user would not like to gaze at the numerous advertisements, logos which are put on a newspaper for a more leisurely reading. Content adaptation steps in at this point where we can aim at extracting only the relevant portions of the HTML code and convert it into WML. This will mean a substantial saving in terms of network bandwidth and resources.

All newspaper sites have a list of all the various sections on the left-hand side of the home-page. This includes sections on sports, entertainment, business etc., which are essentially links to those sections of the newspaper. Content adaptation for newspaper sites can focus on providing only the most important news headlines. Details may be provided only if a client requests for it. Besides, different users have different interests. Hence news items maybe classified according to contents and ordered according to the user's preferences. This means that the WML cards have to be designed and arranged appropriately. Figure 3.5 represents a structure, which can be adopted for serving news to a WAP user.

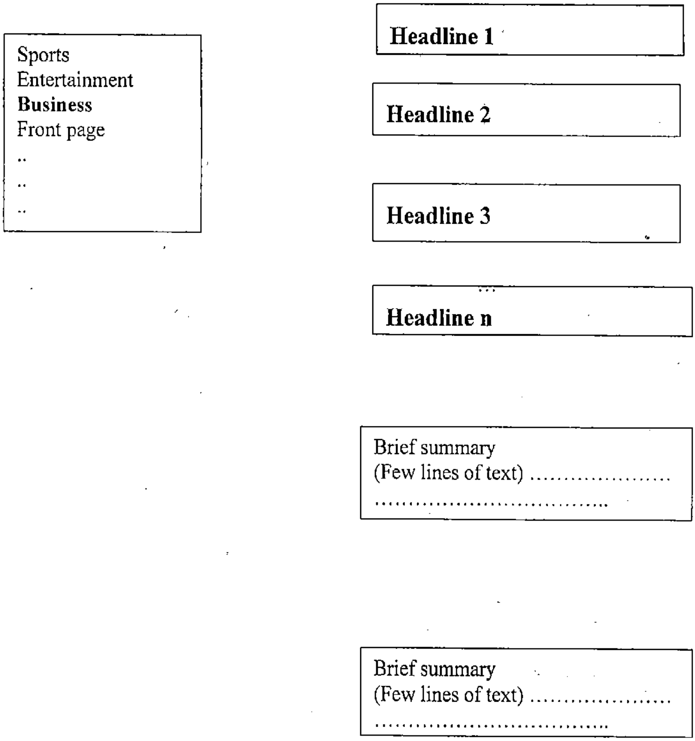


Figure. 3.5 A Typical web site structure

In the following sub-sections we describe the basic framework of our system anc adaptation rules we apply to select the content the from the HTML page to be convertec into WML.

3.2 BASIC FRAMEWORK

The basic framework of our Content Adaptation System is shown in figure. 3.6 The clien (user) request for a URL is relayed to the origin server by the proxy server. The HTMl page is fetched and passed to the Adaptation System. The Adaptation System consists c two main modules:

1. Parser.
2. Wrapper.

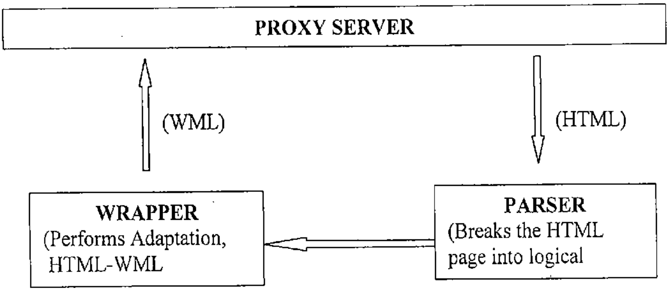


Figure. 3.6

We describe each of these modules in the following sections. The Adaptation Syste uses adaptation rules to select content and performs conversion to WML. The output is set of WML decks corresponding to the requested HTML page. (figure.). A single de is sent to the client. The decks form a logical order and it is possible for the user to navigate through these decks.

Parser

This module scans the HTML page and breaks it into logical units using the tag's name, its attribute and content associated with the tag (e.g. the text occurring between <a> </a>) and plain text. We have found that the most important tags for content adaptation of news items for WAP devices are "<a>" and "<b>" tag.

- <a> : anchor tag. Content occurring between <a> and </a> acts as a link to HTML a page.
- <b>: bold tag. Content occurring between <b> and </b> is rendered in bold.

Tags like <img>, <frames> etc. can be ignored since images do not contain the necessary information which a mobile user accessing news will wish to see.

Thus we can work we a limited set of tags that are picked and stored for converting from HTML to WML. This selected set of tags with its attributes is passed to the Wrapper module.

Wrapper

The input to this module is the portion of the HTML document that the Parser identifies as useful. Essentially, it consists of the text associated to <a> and <b> tags. This module extracts relevant data this text HTML document and uses adaptation rules to determine the content to undergo HTML-to-WML conversion. The Wrapper outputs a set of WML decks that contains the relevant information extracted from the

HTML page requested by the user. For the user to navigate through the decks, it is essential that the cards/decks are in a logical order. It is the aim of the Content Adaptation System to provide a transparent view of the HTML page through this set of WML decks.

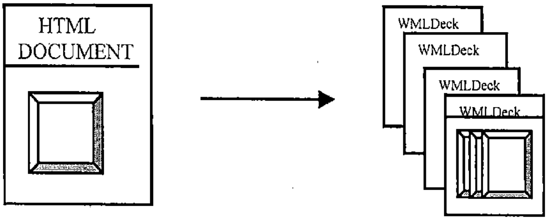


Figure. 3.7 (Output produced by Wrapper module)

3.3 ADAPTATION RULES

The HTML-to-WML conversion is performed by a simple mapping between HTML and WML tags. For example, the tag "<html>" is mapped to the WML tag "<wml>", the "<a>" tag is mapped to "<a>" tag and so on. However, one needs to add a few more conditions at this stage before blindly converting all contents of <a> or <b> tags. For example, all the content that occur between the <a> <a> tags or <b. and </b> tags are not necessarily important. With reference to figure 3.3, there are links to "Feature", "Young World Summer special" etc. which are part of supplements which appear along with the newspaper that do not contain daily news. There are links to editorial columns (not seen in the figure) which is not likely to be read using the mobile device. They would be picked by the Parser module since they occur between, <b> </b> tags. Thus a simple transcoding from HTML-to-WML without adaptation rules being applied is not a good strategy. To determine the utility of content, we make use of keywords that are used to describe the various sections of the newspaper, for example, sports, entertainment, business etc. As seen in figure 3.3 these keywords occur between the <b> </b> tags which would have been picked by the Parser module.

There are also redundancies in an HTML page. A typical site (ref. Figure 3.3) contains information that is redundant, or may not be of primary interest to the user. For example, as seen from figure 3.1, anchors linking to various sections of the newspaper like sports, entertainment, business etc. occur in the top, bottom and on the left side of the page. All these occur between the <a> /<a> tags or <b> </b> tags. They link to the same pages. Showing the same contents in different cards will only frustrate the user. The page also contains many banners, logos and advertisements (ref. Figure 3.2). These elements need to be ignored.

There are certain features specific to a particular site which need to be considered for one to capture all the necessary information. There are some sections of the newspaper which are referred by different names. For example, a link "National" in the *Hindu* website refers to news pertaining to India and a link "International" refers to world news (see figure 3.1). These sections are referred by the *times of india* website by the links "India" and "World" respectively (see figure 3.2). Thus the set of keywords differ from site to site. These keywords are collected by first viewing the sites.

We can describe our adaptation rules as follows:

1. All links which are referred to by the keywords are useful. These are either single

words or compound words. With reference to figure 3.3 this rule is interpreted in the following way. If any of the keywords occurs as <a> *keyword* </a> or <a> <b> *keyword*</b></a> or <b><a> *keyword*<a></b>, it means that they are links to sub-sections and are required.

2. Links, which are headlines and have a brief summary following the headline. These are in the form of a sentence. Figure 3.4 shows an example. The text, which follows the headline is a brief summary of the news headline. The headline is used as a link to a more detailed report. This rule says that if some plain text follows an anchor (<a> tag), and the content of <a> </a> is a sentence, then this content is a headline and the text following the <a> tag is a summary of that headline. Figure gives the snapshot of the page. The items that are selected are also shown.

3.4 Conversion into WML decks

The HTML document is a hypertext which contains links to other documents. Some of these links are present as URL's have global addresses while some of them have relative addresses with respect to the parent document. In news sites most of the links that are relevant for adaptation to WML are relative. This is because mostly the links contain reports while the parent page contains only the headlines.

In the previous section we have described how relevant portion of the text are extracted from the front page of a news site. We have selected those contents, each of which have two basic parts - a headline and a report. In this section we will describe how a logical flow is established between these sets when converted to the WML form. All the content with a headline and a report following it extracted.

The headlines are put together in one single deck. This deck contains a single card. Each of the headline is linked to a separate deck containing the report that followed that headline in the original HTML document. Upon request for a HTML document for a news site, the deck containing the headlines is sent to the user. The user follows the links to read the report that followed that particular headline in the original HTML document if he wishes to do so. The deck containing the report is then sent. In this way, information exchanged between the proxy-server and the user is optimized. Yet the user is able to view the contents of his interest when he wishes to do so.

In this way, we are able to provide a logical view of the most relevant content of the original HTML document to a WAP user.

CHAPTER 4

IMPLEMENTATION AND RESULTS

In this chapter we describe the implementation of the Content Adaptation System for news sites. The basic design of our system is shown in figure 4.1. We have adopted a proxy-based approach for content adaptation. We describe the key components of the system.

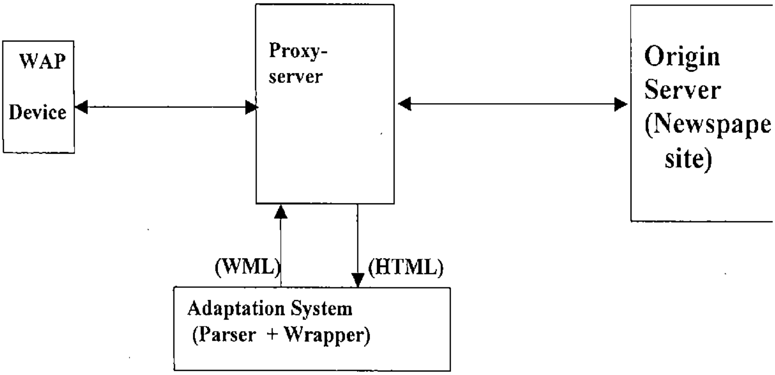


Figure 4.1

4.1 Proxy-server

Our content adaptation system has been implemented into a proxy-server. We have used the public domain Apache server. It was configured to handle WAP related file extensions. This was done by adding the following MIME-types to the MIME types file:

Doc. type	**Official MIME type**	**Extension**
WML document	text/vnd.wap.wml	.wml
WMLScript	text/vnd.wap.wmlscript	.wmls
Compiled WMLScript	application/vnd.wap.wmlscriptc	.wmlsc

4.2 Adaptation System

This consists of the two main modules – Parser and Wrapper.These modules have been mplemented in java (version 1.1). We will give a brief description of the various classes nd methods of these modules.

Class Tags.java.

Constructor Tags().

This class contains the fields required to store information about a HTML tag like its name, attribute, content etc.

ields String name.

Name of the tag.

StringBuffer attribute.

This holds the tag's *attribute*. For example, the anchor tag <a> an attribute "href". The *attribute* name and its value is stored in the attribute field.

StringBuffer content.

This holds the text that is associated with the tag. For example text occurring between <b> and </b> is stored in content field.

StringBuffer morecontent.

This field holds the plain text occurring in the document. It is associated with the <a> tag. This is the brief summary that follows the headline in the HTML page.

Class **Parser.java**

Constructor **Parser(BufferedReader buf).**

Constructs a Paser object to parse the datastream (HTML document) passed from the BufferedReader buf.

Description This class contains methods that scans a HTML document , breaks it into logical units like tag name, tag attribute, tag content.

Fields Vector **validtags**. An array containing the set the of tag names that are to be picked up.

StreamTokenizer **tok**. This object breaks its input into tokens using a syntax table. This syntax table is constructed when the Parser object is instantiated.

Tags[] **arrayoftags**. An array where the tags that are picked up are stored.

Methods **readInput(StreamTokenizer tok)**

This method is called to parse the HTML page. It uses the syntax table

(created when the parser object was instantiated) to recognize the tokens.

Whenever a "<" in encountered, it means start of a tag and "</" is encountered it means the end of a tag.

handlestarttag(StreamTokenizer tok) .

This method is called when "<" is encountered. The next token will be the Tag's name. Different tags are handled differently. For example, the anchor tag (<a>) contains attribute "href" which is to be picked up. The bold tag(<b>) does not have any attribute to be picked up.

handlecomment(StreamTokenizer tok)

This method is called when "<!" is encountered. All characters occurring between "<!" and "-->" need to be ignored. These are the comment characters for a HTML document.

handletext(StreamTokenizer tok)

This method picks the plain text from a html document.

handleignore (StreamTokenizer tok)

This method is called when the start tag of a tag which is to be ignored is encountered. It simply loops through the input till “</” is encountered.

Class **Wrapper.java**

Constructor **Wrapper ().**

Description This class contains methods to extract the relevant information from the text document stored by the Parser. The extraction rules are contained in the methods of this class.

Fields Hashtable **wmlrules.**

This hashtable contains as key-value pairs the HTML tag name and the WML tag to which it is mapped. For example, “<html>” is mapped to ”<wml>” .

Hashtable **dictionary.**

This table contains the list of keywords based on which the relevance of the content is determined. This table is constructed when a wrapper object is instanstiated.

Wmltags[] **wmlarray.**

This holds the WML tags that have been converted from HTML. The class Wmltags is a class similar to the Tags class.

Methods **makewmlcard(Tags[] tagarray)**

This is the main method that converts the HTML tags to WML tags using the wmlrules table. The array "tagarray” contains the set of HTML tags selected by the parser module. It creates the WML decks and cards and arranges them in a logical order such that it is possible to navigate through the decks. It stores the WML tags in the array wmltags and calls the print () method to print the generated WML decks in a set of files.

Print(wmltags[] wmlarray)

This method prints the contents of array 'wmlarray'. 'wmlarray' is an array of WML tags with its attribute and content. This method uses the dictionary of keywords to determine the importance of the content associated with the tag. Each deck is written into a different file.

4.3 SERVLETS

In this we used java servlets to complete HTTP request. Actually we can run these sevlets on any machine.

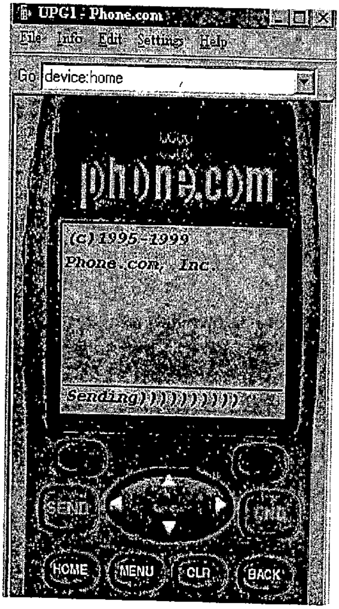
The work of servlet here is to take HTML page and then perform the necessary conversion (HTML to WML) .

The main class used in this is:

Class	**HttpServlet**
Description	This is an abstract class that reside in the javax.servlet.http package. Because it is abstract, it cannot be instantiated. Rather , when building HTTP servlet, we extend the HttpServlet class and implement at least one of its method.

4.4 WAP device

To test the converted WML decks generated by the Adaptation system, we use a WAP device simulator. We are using UP Simulator 4.0 (www.phonc.com). It provides a user interface similar to a WAP phone. The input is through keyboard and mouse. Figure 4.2 shows a snapshot of the simulator. The browser interprets WML and WMLScript.



Go device:home

(c) 1995-1999
Phone.com, Inc.

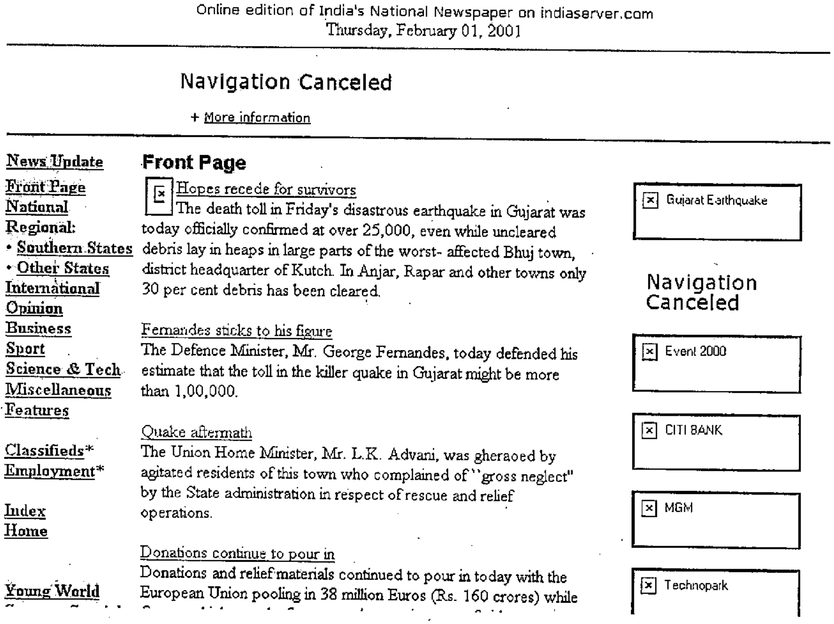
Sending)))))))))))

Figure 4.2

4.5 RESULTS

In this section we present some of the results we have obtained by applying our content adaptation technique on two new site, namely Times of India and Hindu.

Figure 4.3, 4.4, 4.5 show snapshots of the Hindu news site (www.the-hindu.com) and figure 4.6, 4.7 show that of Times of India site (www.timesofindia.com/today/pagehome.htm). The common underlying structure which we had adopted (ref. Figure , chapter 3) is clearly visible. Left-hand side of the page contain links to various sections of the news, the middle portion contain headlines followed by s report on that headline. Our adaptation system is designed to extract these content and convert them into WML decks. The resulting deck containing the headlines is one that is sent to the user first. This deck is shown in figure 4.6. Note that all the headlines do not fit into the screen. One has to scroll down to view the remaining. The word "link" appears on the bottom left of the screen meaning that these headlines lead to another WML deck or card (figure 4.8). The WML source code for each of the decks are also shown. Following a link leads to the report that followed the headline in the original HTML page. (see figure 4.3,4.4,4.5). Thus a user is given the choice to view the report accompanying the headlines only if he wishes to see. The user can return to headline page using the history mechanism of the WML browser.



Online edition of India's National Newspaper on indiaserver.com
Thursday, February 01, 2001

Navigation Canceled

Front Page

☒ Hopes recede for survivors
The death toll in Friday's disastrous earthquake in Gujarat was today officially confirmed at over 25,000, even while uncleared debris lay in heaps in large parts of the worst- affected Bhuj town, district headquarter of Kutch. In Anjar, Rapar and other towns only 30 per cent debris has been cleared.

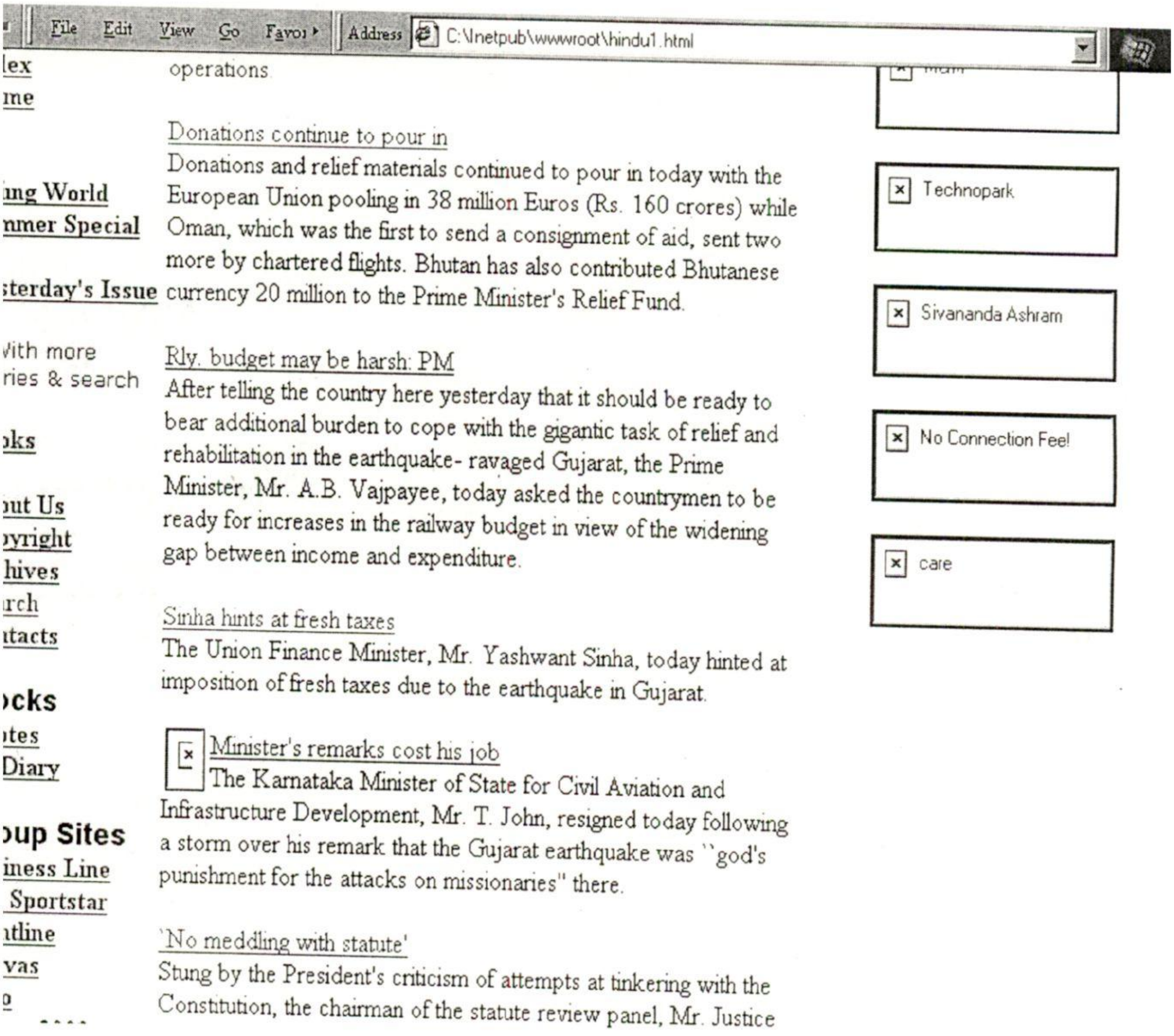
Fernandes sticks to his figure
The Defence Minister, Mr. George Fernandes, today defended his estimate that the toll in the killer quake in Gujarat might be more than 1,00,000.

Quake aftermath
The Union Home Minister, Mr. L.K. Advani, was gheraoed by agitated residents of this town who complained of ''gross neglect" by the State administration in respect of rescue and relief operations.

Donations continue to pour in
Donations and relief materials continued to pour in today with the European Union pooling in 38 million Euros (Rs. 160 crores) while

Figure 4.3

(Shows a snapshot of the Hindu site www.the-hindu.com)



operations

Donations continue to pour in

Donations and relief materials continued to pour in today with the European Union pooling in 38 million Euros (Rs. 160 crores) while Oman, which was the first to send a consignment of aid, sent two more by chartered flights. Bhutan has also contributed Bhutanese currency 20 million to the Prime Minister's Relief Fund.

Rly. budget may be harsh: PM

After telling the country here yesterday that it should be ready to bear additional burden to cope with the gigantic task of relief and rehabilitation in the earthquake- ravaged Gujarat, the Prime Minister, Mr. A.B. Vajpayee, today asked the countrymen to be ready for increases in the railway budget in view of the widening gap between income and expenditure.

Sinha hints at fresh taxes

The Union Finance Minister, Mr. Yashwant Sinha, today hinted at imposition of fresh taxes due to the earthquake in Gujarat.

☒ Minister's remarks cost his job

The Karnataka Minister of State for Civil Aviation and Infrastructure Development, Mr. T. John, resigned today following a storm over his remark that the Gujarat earthquake was ``god's punishment for the attacks on missionaries" there.

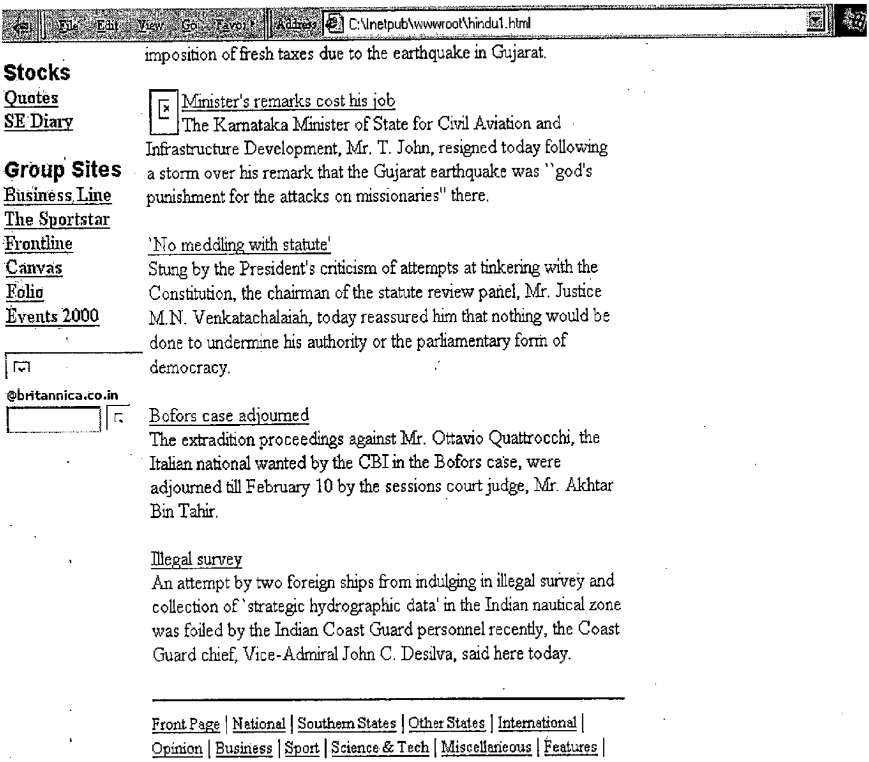
'No meddling with statute'

Stung by the President's criticism of attempts at tinkering with the Constitution, the chairman of the statute review panel, Mr. Justice

Figure 4.4

(Shows a snapshot of the Hindu site www.the-hindu.com)

510648.



imposition of fresh taxes due to the earthquake in Gujarat.

Stocks
Quotes
SE Diary

Group Sites
Business Line
The Sportstar
Frontline
Canvas
Folio
Events 2000

@britannica.co.in

Minister's remarks cost his job
The Karnataka Minister of State for Civil Aviation and Infrastructure Development, Mr. T. John, resigned today following a storm over his remark that the Gujarat earthquake was "god's punishment for the attacks on missionaries" there.

'No meddling with statute'
Stung by the President's criticism of attempts at tinkering with the Constitution, the chairman of the statute review panel, Mr. Justice M.N. Venkatachalaiah, today reassured him that nothing would be done to undermine his authority or the parliamentary form of democracy.

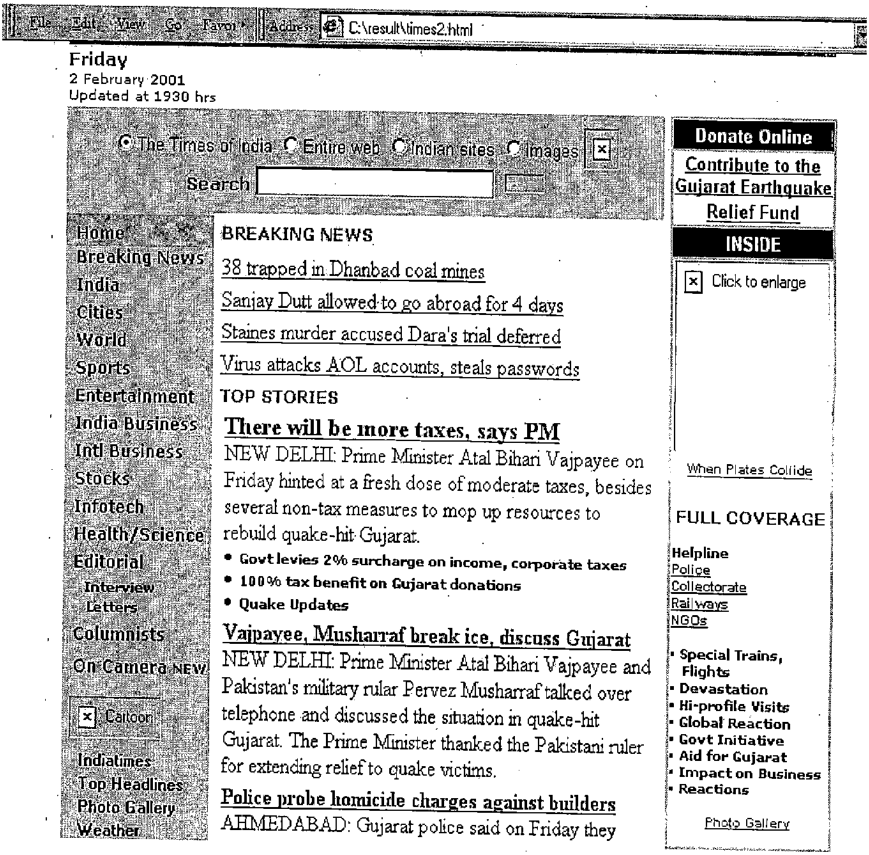
Bofors case adjourned
The extradition proceedings against Mr. Ottavio Quattrocchi, the Italian national wanted by the CBI in the Bofors case, were adjourned till February 10 by the sessions court judge, Mr. Akhtar Bin Tahir.

Illegal survey
An attempt by two foreign ships from indulging in illegal survey and collection of 'strategic hydrographic data' in the Indian nautical zone was foiled by the Indian Coast Guard personnel recently, the Coast Guard chief, Vice-Admiral John C. Desilva, said here today.

Front Page | National | Southern States | Other States | International | Opinion | Business | Sport | Science & Tech | Miscellaneous | Features |

Figure 4.5

(Shows a snapshot of the Hindu site www.the-hindu.com)



Friday
2 February 2001
Updated at 1930 hrs

Contribute to the Gujarat Earthquake Relief Fund

BREAKING NEWS

38 trapped in Dhanbad coal mines

Sanjay Dutt allowed to go abroad for 4 days

Staines murder accused Dara's trial deferred

Virus attacks AOL accounts, steals passwords

TOP STORIES

There will be more taxes, says PM

NEW DELHI: Prime Minister Atal Bihari Vajpayee on Friday hinted at a fresh dose of moderate taxes, besides several non-tax measures to mop up resources to rebuild quake-hit Gujarat.

- Govt levies 2% surcharge on income, corporate taxes
- 100% tax benefit on Gujarat donations
- Quake Updates

Vajpayee, Musharraf break ice, discuss Gujarat

NEW DELHI: Prime Minister Atal Bihari Vajpayee and Pakistan's military rular Pervez Musharraf talked over telephone and discussed the situation in quake-hit Gujarat. The Prime Minister thanked the Pakistani ruler for extending relief to quake victims.

Police probe homicide charges against builders

AHMEDABAD: Gujarat police said on Friday they

FULL COVERAGE

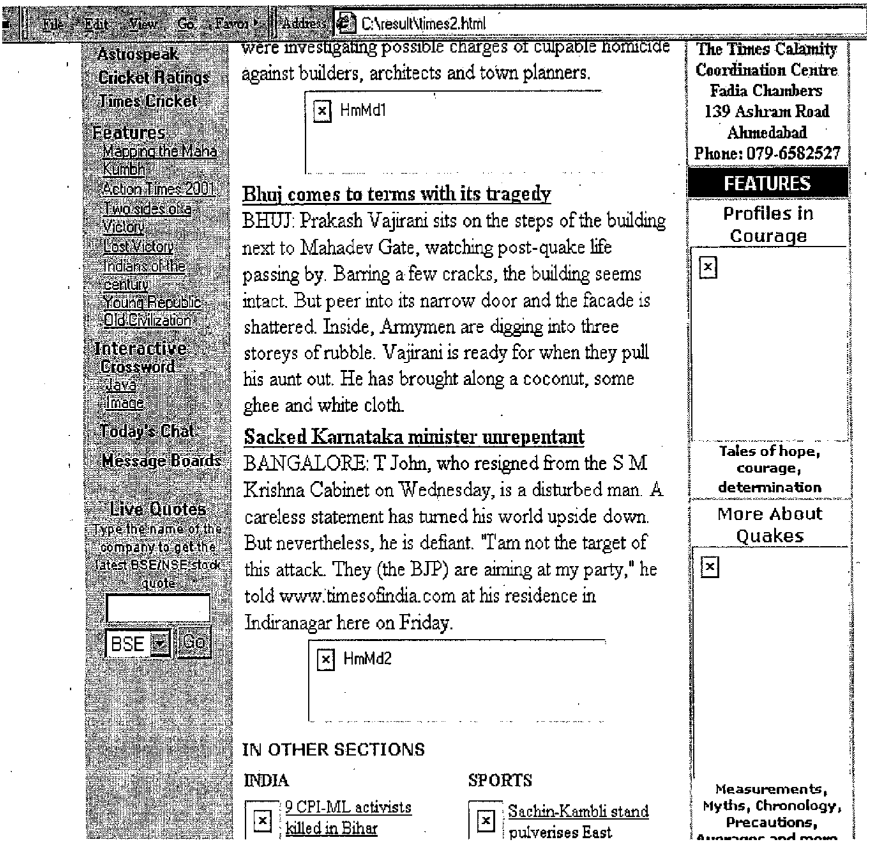
Helpline
Police
Collectorate
Railways
NGOs

- Special Trains, Flights
- Devastation
- Hi-profile Visits
- Global Reaction
- Govt Initiative
- Aid for Gujarat
- Impact on Business
- Reactions

Figure 4.6

(Snapshot showing part of the times of India site

www.timesofindia.com/today/pagehome.htm)



were investigating possible charges of culpable homicide against builders, architects and town planners.

Bhuj comes to terms with its tragedy

BHUJ: Prakash Vajirani sits on the steps of the building next to Mahadev Gate, watching post-quake life passing by. Barring a few cracks, the building seems intact. But peer into its narrow door and the facade is shattered. Inside, Armymen are digging into three storeys of rubble. Vajirani is ready for when they pull his aunt out. He has brought along a coconut, some ghee and white cloth.

Sacked Karnataka minister unrepentant

BANGALORE: T John, who resigned from the S M Krishna Cabinet on Wednesday, is a disturbed man. A careless statement has turned his world upside down. But nevertheless, he is defiant. "I am not the target of this attack. They (the BJP) are aiming at my party," he told www.timesofindia.com at his residence in Indiranagar here on Friday.

IN OTHER SECTIONS

INDIA

9 CPI-ML activists killed in Bihar

SPORTS

Sachin-Kambli stand pulverises East

The Times Calamity Coordination Centre
Fadia Chambers
139 Ashram Road
Ahmedabad
Phone: 079-6582527

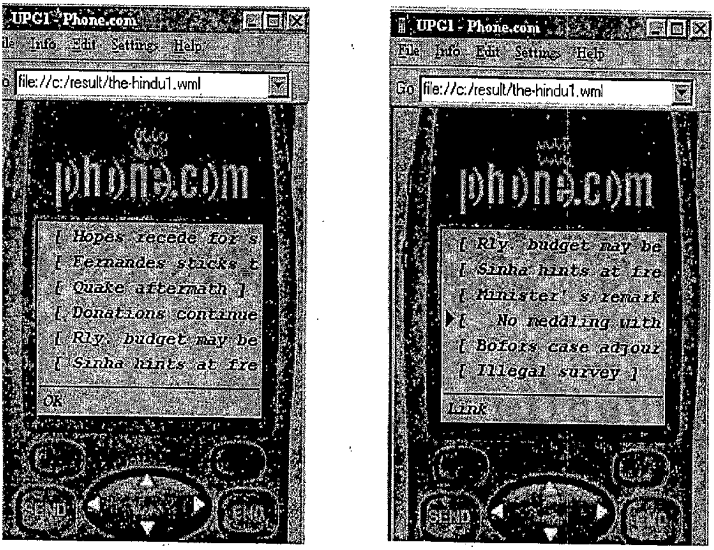
Tales of hope, courage, determination

Measurements, Myths, Chronology, Precautions, Averages and more

Figure 4.7

(Snapshot showing part of the Times of India site)

Figure 4.8 shows www.the-hindu.com as seen in up.sdk simulator.



file://c:/result/the-hindu1.wml

[Hopes recede for s
[Fernandes sticks t
[Quake aftermath]
[Donations continue
[Rly. budget may be
[Sinha hints at fre

file://c:/result/the-hindu1.wml

[Rly. budget may be
[Sinha hints at fre
[Minister' s remark
[No meddling with
[Bofors case adjour
[Illegal survey]

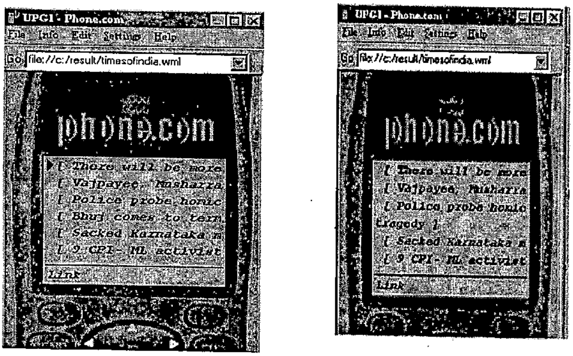
Figure 4.8

(snapshot of up.sdk simulator displaying www.the-hindu.com site)

The WML source code of the deck generated for the Hindu news site (figure 4.8).

```
<?xml version="1.0"?>
<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">
<wml>
<card  id=" cd1">
<p  mode="nowrap">
<a href = "http://127.0.0.1:8080/the-hindu1.wml/#cd25">
 Hopes recede for survivors
</a>
<a href = "http://127.0.0.1:8080/the-hindu2.wml/#cd26">
 Fernandes sticks to his figure
</a>
<a href = "http://127.0.0.1:8080/the-hindu3.wml/#cd27">
 Quake aftermath
</a>
<a href = "http://127.0.0.1:8080/the-hindu4.wml/#cd28">
 Donations continue to pour in
</a>
<a href = "http://127.0.0.1:8080/the-hindu5.wml/#cd29">
 Rly. budget may be harsh: PM
</a>
<a href = "http://127.0.0.1:8080/the-hindu6.wml/#cd210">
 Sinha hints at fresh taxes
</a>
<a href = "http://127.0.0.1:8080/the-hindu7.wml/#cd211">
 Minister' s remarks cost his job
</a>
<a href = "http://127.0.0.1:8080/the-hindu8.wml/#cd212">
` No meddling with statute'
</a>
<a href = "http://127.0.0.1:8080/the-hindu9.wml/#cd213">
 Bofors case adjourned
</a>
<a href = "http://127.0.0.1:8080/the-hindu10.wml/#cd214">
 Illegal survey
</a>
</p>
</card>
</wml>
```

Figure 4.9 shows times of India site seen in up.sdk simulator



file://c:/result/timesofindia.wml

[There will be more
[Vajpayee, Musharra
[Police probe homic
[Bhuj comes to term
[Sacked Karnataka m
[9 CPI- ML activist

Link

(a)

file://c:/result/timesofindia.wml

[There will be more
[Vajpayee, Musharra
[Police probe homic
tragedy]
[Sacked Karnataka m
[9 CPI- ML activist

Link

(b)

Figure 4.9

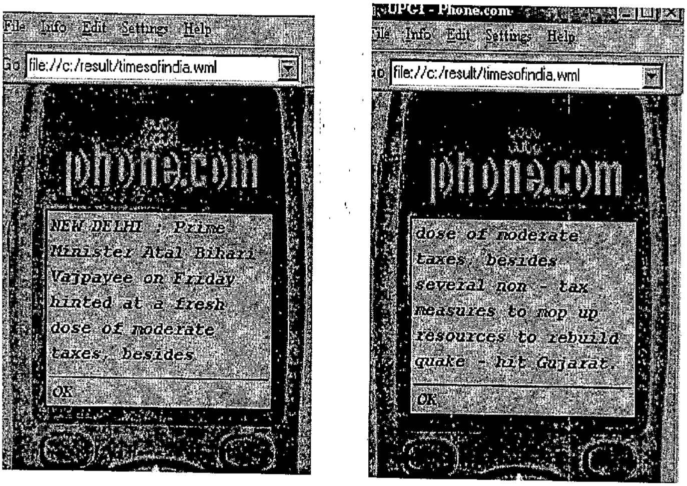
(snapshot of up.sdk simulator displaying www.timesofindia.com site)

The WML source code of the deck generated for the Times of India news site (fig 4.9).

```
<?xml version="1.0"?>

<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>

<card  id=" cd1">

<p  mode="nowrap">

<a href = "http://127.0.0.1:8080/timesofindia1.wml/#cd25">

 There will be more taxes, says PM

</a>

 <a href = "http://127.0.0.1:8080/timesofindia2.wml/#cd26">

 Vajpayee, Musharraf break ice, discuss Gujarat

</a>

  <a href = "http://127.0.0.1:8080/timesofindia3.wml/#cd27">

 Police probe homicide charges against builders

</a>

  <a href = "http://127.0.0.1:8080/timesofindia4.wml/#cd28">

 Bhuj comes to terms with its tragedy

</a>

  <a href = "http://127.0.0.1:8080/timesofindia5.wml/#cd29">

 Sacked Karnataka minister unrepentant

</a>

 <a href = "http://127.0.0.1:8080/timesofindia6.wml/#cd210">

9 CPI- ML activists killed in Bihar gangwar

</a>
```

```
</p>
</card>
</wml>
```

The link of "There will be more taxes, says PM" is shown in figure 4.10



file://c:/result/timesofindia.wml

NEW DELHI : Prime Minister Atal Bihari Vajpayee on Friday hinted at a fresh dose of moderate taxes, besides

file://c:/result/timesofindia.wml

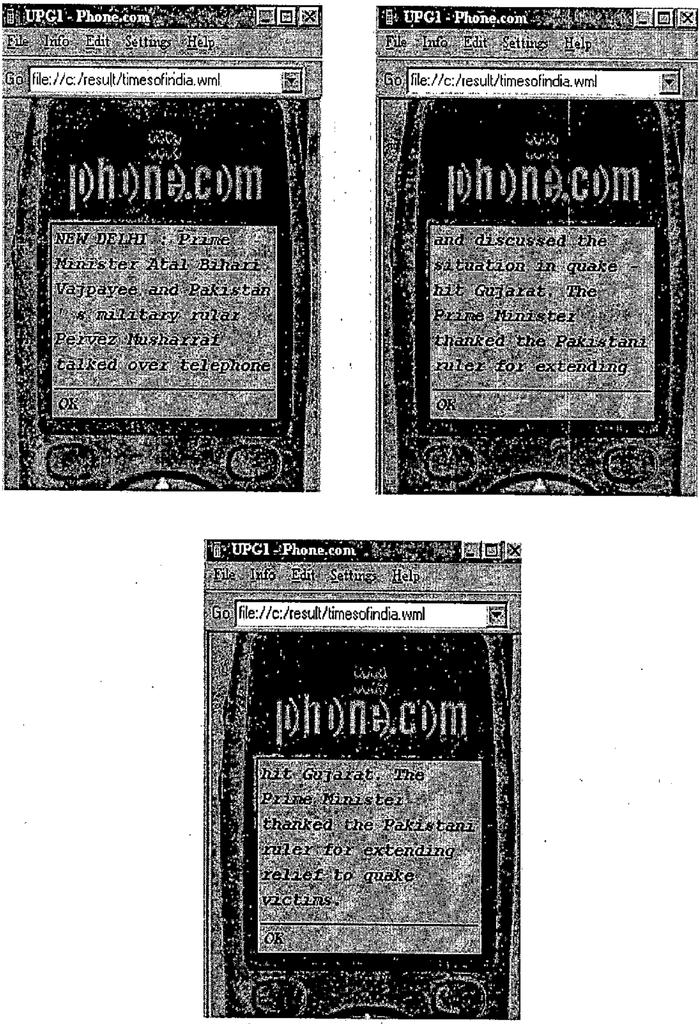
dose of moderate taxes, besides several non - tax measures to mop up resources to rebuild quake - hit Gujarat.

Figure 4.10

The WML source code for figure 4.10 is

```
<?xml version="1.0"?>

<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>

<card  id="cd25">

<p>

 NEW DELHI : Prime Minister Atal Bihari Vajpayee on Friday hinted at a fresh dose of
moderate taxes, besides several non - tax measures to mop up resources to rebuild quake -
hit Gujarat.

</p>

</card>

</wml>
```

The link of “Vajpayee,Musharraf break ice,discuss Gujarat ” is shown in figure 4.11



file://c:/result/timesofindia.wml

NEW DELHI : Prime
Minister Atal Bihari
Vajpayee and Pakistan
's military ruler
Pervez Musharraf
talked over telephone

OK

file://c:/result/timesofindia.wml

and discussed the
situation in quake -
hit Gujarat. The
Prime Minister
thanked the Pakistani
ruler for extending

OK

file://c:/result/timesofindia.wml

hit Gujarat. The
Prime Minister
thanked the Pakistani
ruler for extending
relief to quake
victims.

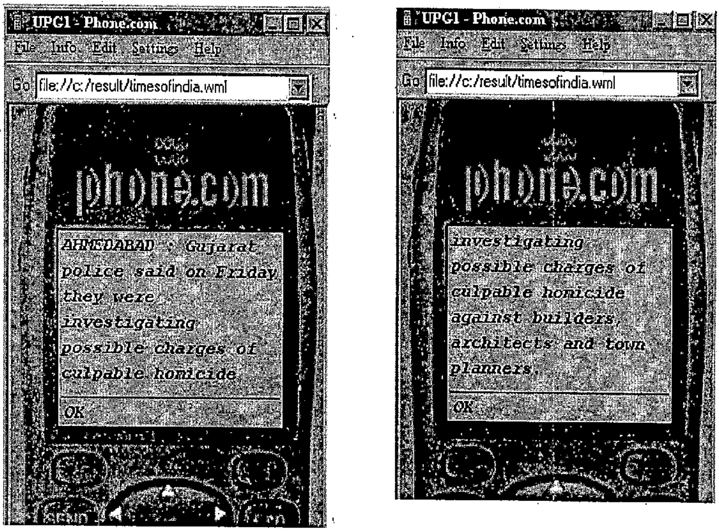
OK

Figure 4.11

The WML source code for figure 4.11 is

```
<?xml version="1.0"?>

<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>

<card id="cd26">

<p>

 NEW DELHI : Prime Minister Atal Bihari Vajpayee and Pakistan ' s military rular
Pervez Musharraf talked over telephone and discussed the situation in quake - hit Gujarat.
The Prime Minister thanked the Pakistani ruler for extending relief to quake victims.

</p>

</card>

</wml>
```

The link of "Police probe homicide charges against builders " is shown in figure 4.12


UPG1 - Phone.com
File Info Edit Settings Help
Go file://c:/result/timesofindia.wml
phone.com

AHMEDABAD : Gujarat police said on Friday they were investigating possible charges of culpable homicide

OK

UPG1 - Phone.com
File Info Edit Settings Help
Go file://c:/result/timesofindia.wml
phone.com

investigating possible charges of culpable homicide against builders, architects and town planners.

OK

Figure 4.12

The WML source code for figure 4.12 is

```
<?xml version="1.0"?>

<!DOCTYPE wml PUBLIC "-//WAPFORUM//DTD WML 1.1//EN"
"http://www.wapforum.org/DTD/wml_1.1.xml">

<wml>

<card  id="cd27">

<p>

 AHMEDABAD : Gujarat police said on Friday they were investigating possible charges
of culpable homicide against builders, architects and town planners.

</p>

</card>

</wml>
```

It is clear from the results that our content adaptation system is able to extract the most relevant information from the news sites and able to present the converted WML decks in a logical manner.

Chapter 5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

We have developed a content adaptation system that can automatically create a WML document from HTML documents. It is done by extracting certain information from the HTML documents. The most obvious concern for this kind of application is to decide which parts of the HTML document should be adapted for the WAP device. This is particularly important keeping in view the resource limitations of the WAP devices.

Ideally, a translation system which can understand content and extract important portions of text appropriately would be perfect solution to this problem. However with the current state of natural language understanding it is not possible to do so. Hence it is required to do a domain based analysis for content adaptation.

Our system focuses on doing a domain dependent content adaptation from HTML documents for WAP devices. However, while considering the domain we do not depend on any natural language attribute but only the structure of HTML documents. Our specific application is for translating news site contents suitably for a WAP device. This is an important area from the applications point of view since news is one of the most important content which a mobile user would like to receive through the Internet.

These web sites have a fixed underlying structure that was exploited to develop rules for extracting the most relevant information from the HTML documents. This increases the efficiency of the adaptation system and the rules formulated apply to large set of dynamically changing web documents. This underlying structure does not change and rules once formulated will work in future too.

There is a certain commonality among all news sites through they are not exactly same. A content adaptation system developed for one news site can be easily modified to suit the requirements of other news sites. The adaptation rules for a site can be stored as meta-knowledge associated to that site which is used by the proxy-server for conversion of HTML documents of that site to WML form. We have illustrated the functioning of our system through some results obtained over different news sites-predominantly maintained by leading newspaper houses of India. Through our approach we have been able to provide the mobile user a transparent view of their familiar web sites.

5.2 Future work

With the explosive growth of the Internet, more and more information Information is available on the Internet and mobile users will also need to access it. Content adaptation provides a solution for mobile users to access the already existing information. This is a very active field of research since it makes it possible for a larger audience to access the same content. The content adaptation technique relies on the common structure of the web documents. It would be desirable do develop an interface to modify the extraction rules according to one's preferences. Future works can focus on developing authoring tools that allow one to write extraction rules for domain specific applications.

REFERENCES

1. Kylanapaa M. and Laakko T. - Adapting Content to Mobile Terminals: Examining Two Approaches, Third Generation Mobile System in Europe, London 25-27.1.1999.

2. http://www.wapforum.org/what/technical.htm

3. http://www.wapforum.org

4. W3C XML Recommendation 1998. http://www.w3c.org/XML

5. "WML Specification",WAP Forum, 04-November-1999.
   URL: http://www.wapforum.org/

6. "WML Script Specification",WAP Forum, 04-November-1999.
   URL: http://www.wapforum.org/

7. Rakesh Mohan, John R. Smith, " Adapting Multimedia Internet Content for Universal Access, IEEE Transactions on Multimedia," Vol. 1, No.1, March 1999, pg 104-114.

8. C-S Li, R. Mohan and John R. Smith, "Multimedia Content Description in the Infopyramid," Proc. ICASSP'98 , Special Session on Signal Processing in Modern Multimedia Standard, Seatle, WA , USA.

9. Real Networks http://www.real.com

10. Intel's quickweb http://www.intel.com/quickweb

11. Proxinet http://www.proxinet.com

12. T. Bickmore and B. Schilit, "Digestor: Device Independent Access to the World Wide Web", proceedings of the Sixth International World Wide Web conference, Santa Clara, California, 1999

13. Eija Kaasinen, Matti Aaltonen, Juha Kolari, Suvi Melakoski, Timo Laakko: "Two Approaches to Bringing Internet Services to WAP Devices,"
IEEE Transaction on multimedia, Vol 1 , No. 1 , March 1999 , pg.104 – 114.

14. Wei-Ying Ma, Ilja Bedner, Grace Chang, Allan Kuchinsky, and HongJiang Zjang
http://www.cooltown.hp.com/papers/MMCN2000.htm

```
import javax.servlet.*;//using servlets
import javax.servlet.http.*;//for http post and get methods
import java.io.*; //println etc
import java.lang.*;
import java.net.*; //URL class
import java.util.*;


public class testServlet4 extends HttpServlet  {
String[] arry = new String[10];
 int p =0;
url u = new url();
  BufferedReader buf ;



  //http get method
 public void doGet(HttpServletRequest req,HttpServletResponse res) throws
IOException
 {
  String line = "yes..";
    String value="h";
      String name = "y";

  res.setContentType("text/vnd.wap.wml"); //mime format specification
  PrintWriter out = res.getWriter();
  Enumeration e = req.getParameterNames();
      while(e.hasMoreElements())
       name = (String)e.nextElement();
       //value is now hold the url value
       value = req.getParameterValues(name)[0];
String Header="<?xml version=\"1.0\"?>";
      String head = "<!DOCTYPE wml PUBLIC \"-//WAPFORUM//DTD WML 1.1//EN\"
\"http://www.wapforum.org/DTD/wml_1.1.xml\">";

 //now if name is same "y" which we give initially
 if(name.equals("y")){
 buf = new BufferedReader(new FileReader("c:\\jsdk2.1\\examples\\web-
inf\\servlets\\options.wml"));
}// end if
 else

      {
      //u is an object of class url and we are passing the value which
      //contain the url
      u.getPage(value);
      buf = new BufferedReader(new FileReader(u.baseaddr+".wml"));
      if(buf.readLine()==null){
   out.println(Header);
   out.println(head);
   out.println("<wml><card><p>error reading file </p>");
   out.println("</card></wml>");
   }


      }
while((line=buf.readLine())!=null)
```

```
.println(line);

/end method doget
it is http post method
olic void doPost(HttpServletRequest request,
               HttpServletResponse response)
   throws IOException, ServletException
 {
   doGet(request, response);
 }



this is class url which have getpage method which take url as input
nd return uu a object of class URL(inbuiltclass)*/
lic class url {
ing baseaddr;
lic void getPage(String args) {
 URL uu;
   Tags[] arrayoftags=new Tags[1000];
 try{
    uu = new URL("http://"+args);//args passed from "value"
 //buf contain the html page read using bufferreader
 BufferedReader  buf = new BufferedReader( new
utStreamReader(uu.openStream()));
 Parser pas = new Parser(buf);
 pas.readInput(arrayoftags);
   buf.close();
eaddr=args;
aseaddr=baseaddr = baseaddr.substring(4,baseaddr.indexOf('.',4));;
   Wrapper wm = new Wrapper();
   wm.makewmlcard(arrayoftags,pas.pointer);

  }catch(MalformedURLException e){
   System.out.println("URL MUST BE OF THE FORM http://www.yahoo.com");
    }
   catch(IOException f){}

' end method get




s Parser

t counter;
leOutputStream fil; //output steram that writes to the file
intWriter outfl; //output stream that contain print() and println()
ack tags; //tags is now a stack
ctor validtags; // contains the set of tags to be picked
l v; //url class defined in program
gs[] arrayoftags; // array in which the tags that are picked are stored
t pointer;
reamTokenizer tok;

//parser is taking the buf as input"html"
this class parse the html documents and seprate the valid tags
```

```
//the main tags which are required are <a></a><b></b><href....>
Parser(BufferedReader buf) throws IOException
  {
    tags = new Stack();
    /*steramTokenizer is inbuilt class
    this object breaksits input into tokens using syntax table
    this syntax table is constructed when the parser object is
    instantiated*/
    tok = new StreamTokenizer(buf);
    //resetSyntax method is to reset default set delimeters
    tok.resetSyntax();
    //whitespaceChar specified white space char
    tok.whitespaceChars(0,' ');
    //this method used to specify the range of valid char from a to z
    tok.wordChars('a','z');
    tok.wordChars('A','Z');
    tok.wordChars('!','!');
    tok.wordChars('&','&');
    tok.wordChars('#','#');
    tok.ordinaryChar(';');
    tok.wordChars('.','.');
    tok.wordChars(',',',');
    tok.ordinaryChar('<');
    tok.ordinaryChar('>');
    tok.ordinaryChar('=');
    tok.ordinaryChar('/');
    tok.ordinaryChar('-');
     System.out.println("constructed..");
   //this is for seprate the valid tags
    validtags = new Vector(0,1); //validtags is a dynamic array
    //addElements(element) add element to vector
    validtags.addElement("html");
    validtags.addElement("body");
    validtags.addElement("a");
    validtags.addElement("b");
    validtags.addElement("br");
    validtags.addElement("font");
    validtags.addElement("h");
    validtags.addElement("h2");
    validtags.addElement("h3");
    validtags.addElement("h4");
    validtags.addElement("h5");
    validtags.addElement("h6");
    validtags.addElement("img");
  }


/*this method is called to parse the html document, it uses syntax table
(created when the parser object was instantiated)to recognize the tokens.
 whenever a "<" is encountered, it means start of tag and "</"
 is encoured it means end of a tag */
void readInput(StreamTokenizer tok) throws IOException
  {
  int  prev=0;
  String strg1="";
  String strg2="";
String strg3="";
String strg4="";
int t=0;
String prevtok="";
 System.out.println("reached....");
while((t=tok.nextToken())!=tok.TT_EOF||counter < 6)
  {
  switch(t) {
  case '<' : strg4 =prevtok="<";
         break;
  case '/' :  strg4 = "/";
              break;
  case -3 : strg3=tok.sval;
               System.out.println(strg3 + "  ---strg--  " +strg4);
    if(strg4.equals("/"))//this is for end tag
     {
       if(this.validtags.contains(tok.sval.toLowerCase()))
         {
         handleendtag(tok);
         strg4=prevtok="";
         strg2="";
         }
       strg4="";
       break;
     }

   if(strg4.equals("<"))
     {
     if(strg3.equals("!"))
       {
        handlecomment(tok); //this is for comment
        strg4="";
        break;
       }
     if(validtags.contains(tok.sval.toLowerCase()))
       {

        handlestarttag(tok); //< start tag
        strg4="";
        break;
       }
     }

 //this is to check that is it text part
  if( (!strg4.equals("<"))&&(!strg4.equals("/")) )
    if(arrayoftags[pointer-1].name.equals("a"))
      if(arrayoftags[pointer-1].content.length()>1)
        {
        tok.pushBack();
        handletext(tok,pointer-1);
        break;
        }
      else
        if(arrayoftags[pointer-2].name.equals("a")&&(arrayoftags[pointer-
ontent.length()>1) )
        {
          System.out.println("from p - 2 ****** "+tok.sval);
```

```
             tok.pushBack();
             handletext(tok,pointer-2);
             break;
           }

     ignoretag(tok);

    default :    break;

    }
    }

  } // end method

void getTitle(StreamTokenizer tok) throws IOException
  {
    System.out.println("in here");
    outf1.println("<TITLE>");
    int i=0;
    do
    {
      switch(i= tok.nextToken())
        {
        case -3 :   outf1.print(tok.sval+" " );
                       break;
        default :
                       break;
        }

    }while(i!='/');
    outf1.println("</TITLE>");
    tok.nextToken();
  } //end method

void getBold(StreamTokenizer tok) throws IOException
  {
    String tag1 = tok.sval.toLowerCase();
    tags.push(tag1);
    Tags tagb = new Tags(tag1);
    int i=0;
    do
    {
      switch(i= tok.nextToken())
        {
        case '<'     :    break;

        case ';'     :
                      tagb.content.append(';');
                      break;
        case '-'     :
                      tagb.content.append('-');
                      break;
        case -3 :                   if(tok.sval.equals("&nbsp"))
                         tok.nextToken();
                        else
                          {
                              tagb.content.append(" "+tok.sval);
                            }
                         break;

          case '>' :               break;

          default :
                        tagb.content.append((char)i);
                        break;
          }

       }while(i!='<');

       tok.pushBack();
       arrayoftags[pointer++]=tagb;
       return;
    }// end method
   /to read href by finding "a"



   'this method is called when "<" is encountered. the next token will be
   ne tag's name. different tag are handeled differently. for eg. the
   nchor tag<a>contains attribute "href"which is to be picked up
   ne bold tag<b> does not have any attribute to be picked up */
   oid handlestarttag(StreamTokenizer tok)throws IOException
    {
      String thistok = tok.sval.toLowerCase();

      if(thistok.equals("br"))
      {
        return;
      }

      if(thistok.equals("title"))
      {
        return;
      }
      if(thistok.equals("h"))
      {
        System.out.println(" in  "+ thistok);
        getBold(tok);
        return;
      }
      if(thistok.equals("h1")||thistok.equals("h2")||thistok.equals("b"))
      {
        getBold(tok);
        return;
      }

      if(thistok.equals("a"))
      {
        tags.push("a");
        Tags taga=new Tags("a");
        int t=0;
        do
          {
```

```
    switch(t=tok.nextToken())
    {
    case ';'    :
                   taga.attribute.append(';');
                   break;
    case -3 :
             if(tok.sval.equals("HREF")||tok.sval.equals("href"))
                       {
                      taga.attribute.append(" "+tok.sval.toLowerCase(
                       }

                    else
                       {
                      taga.attribute.append(" "+tok.sval);
                       }
                    break;
    case '"'    :
                   taga.attribute.append('"');
                   break;
    case '\'' :
                   taga.attribute.append("'");
                   break;
    case '>' :      break;

    default :
                taga.attribute.append((char)t);
                break;

    }
  }while(t!='>');

if(.(tok.nextToken()=='<'))
  {
    tok.pushBack();
    arrayoftags[pointer++]=taga;
    return;
  }
else {
  tok.pushBack();
  do
    {
    switch(t=tok.nextToken())
      {
      case -3 :
        if(tok.sval.equals("&nbsp")||tok.sval.equals("&"))
          tok.nextToken();
        else
          {
          taga.content.append(" "+tok.sval);
          }
        break;
      case '<':          break;

      case ';' :
                     taga.content.append(';');
                     break;
```

```
           default :
                       taga.content.append((char)t);
                       break;

           }
         }while(t!='<');
       tok.pushBack();
     }
     arrayoftags[pointer++]=taga;
     return;
   }

   if(thistok.equals("img"))
   {

     ignoretag(tok);
     return;
   }

   if(thistok.equals("font"))
   {
     ignoretag(tok);
     return;
   }
   if(thistok.equals("html"))
   {
     tags.push("html");
     Tags tagh= new Tags("html");
     arrayoftags[pointer++]=tagh;
     return;
   }

   if(thistok.equals("body"))
   {
     tags.push("body");
     Tags tagbd = new Tags("body");
     arrayoftags[pointer++]=tagbd;
     return;
   }
  return;
}// end method handlestarttag()



is method picks up the palin text from the html document */
id handletext(StreamTokenizer tok,int p)throws IOException
{
  int t=0;
  StringBuffer morecontent = new StringBuffer(1);
  boolean prevword = false;
  do
  {
    switch(t=tok.nextToken())
      {
      case -3 :
                   if(tok.sval.equals("&nbsp"))
                      tok.nextToken();
                      else
                        {
                          prevword = true;
                          morecontent.append(" "+tok.sval);
                        }
                      break;
        case '<':                     prevword=false;
                        break;

        case ';' :
                      morecontent.append(';');
                       break;
        default :      if(prevword)
                    {
                    morecontent.append(" "+(char)t);
                    }
                    else
                      morecontent.append((char)t);
                    prevword = false;
                    break;

        }
    }while(t!='<');
    tok.pushBack();
    if(morecontent.length()>25)
    {
      arrayoftags[p].morecontent=morecontent;
      counter++;
    }
    return;
  }// end method
void handlestarttag(String stg)
  {
    tags.push(stg);
    return;
  }

void handleendtag(String stg)
  {
    if(!tags.empty())
    if(tags.peek().equals(stg))
    ;
    else
    return;
  }

void handleendtag(StreamTokenizer tok)
  {
    String endtag = tok.sval.toLowerCase();
    if(!tags.empty())
    if(tags.peek().equals(endtag)){
    }
    else
    return;
  }

void ignoretag(StreamTokenizer tok) throws IOException
```

```
{
while(tok.nextToken()!='>')
  ;
return;

:his token is called when "<! is encountered. all characters occuring
.ween "<!"and -->"need to be ignored. these are comment in html
:ument */
·id handlecomment(StreamTokenizer tok) throws IOException
{
  char prev=(char)tok.nextToken();
  char next=(char)tok.nextToken();
  while(true)
  {
    if((prev=='-')&&(next=='>'))
      break;
    else
      {
  f     int t=0;
        switch(t=tok.nextToken())
        {
        case -3 : break;
        case ';'          : break;
        case '>'          : if(prev=='-')
                        next='>';
                      break;
        default            :prev=(char)t;

        }
      }
  }
  return;

// end method



d readInput(Tags[] tags) throws IOException

 arrayoftags = tags;
 int  prev=0;
 String strg1="";
 String strg2="";
 String strg3="";
 String strg4="";
 int t=0;
 String prevtok="";
 while((t=tok.nextToken())!=tok.TT_EOF||counter < 6)
 {
   switch(t) {
   case '<' : strg4 =prevtok="<";
     break;
   case '/' :  strg4 = "/";
     break;
   case -3 : strg3=tok.sval;
     if(strg4.equals("/"))
       {
               if(this.validtags.contains(tok.sval.toLowerCase()))
                 {
                   handleendtag(tok);
                   strg4=prevtok="";
                   strg2="";
                 }
              strg4="";
              break;
              }

            if(strg4.equals("<"))
              {
              if(strg3.equals("!"))
                {
                  handlecomment(tok);
                  strg4="";
                  break;
                }
              if(validtags.contains(tok.sval.toLowerCase()))
                {

                  handlestarttag(tok);
                  strg4="";
                  break;
                }
              }
        if( (!strg4.equals("<"))&&(!strg4.equals("/")) )
          if(arrayoftags[pointer-1].name.equals("a"))
            if(arrayoftags[pointer-1].content.length()>1)
              {
              tok.pushBack();
              handletext(tok,pointer-1);
              break;
              }
            else
              if(arrayoftags[pointer-2].name.equals("a")&&(arrayoftags[pointer
 2].content.length()>1) )
              {
                System.out.println("from p - 2 ****** "+tok.sval);
                tok.pushBack();
                handletext(tok,pointer-2);
                break;
              }

        ignoretag(tok);

       default :    break;

       }
       }

    } // end method

} //**************** end class parser *************************

public  class Tags{
    String name;//name of the tag
 StringBuffer attribute;//attribute name
 StringBuffer content; //hold text associated with<b></b>
 StringBuffer morecontent;//hold text associated with <a>tag
 //which is summery  or say headlines
 Tags(String nam,StringBuffer Attribute,StringBuffer Content ){
 name=nam;
:ribute=new StringBuffer();
:ent=new StringBuffer();
 attribute=Attribute;
 content=Content;
 }
ıgs(String Name){
ıe=Name;
:ribute=new StringBuffer(1);
.ent=new StringBuffer(1);
:content=new StringBuffer(1);

*****end class tags **********
.is class contains method to extract the relevent information
. the text document stored by parser. the extraction rules are
ained in the methods of this class. */
s Wrapper


ashtable wmlrules; //table that maps html tag to wml.
ashtable wmlendrules;
ashtable dictionary; // dictionary containing keywords
t CARD_NO = 1, DECK_NO = 1,buffersize =0,start;
ack stck;
t anchorcard=5;
ltags[] wmlarray;
ltags[] wmlarray1;
t pointer=0,pointer1=0;
leOutputStream  wmlfil;
intWriter outwml;
ivate class wmltags

String name;
StringBuffer attribute;
StringBuffer content;
StringBuffer morecontent;
.nt deck_no;
vmltags(String nam,StringBuffer Attribute,StringBuffer Content, StringBuffer
:ontent )
  {
  name=nam;
  attribute=new StringBuffer();
  content=new StringBuffer();
  morecontent=new StringBuffer();
  attribute=Attribute;
  content=Content;
  morecontent=Morecontent;
  }
mltags(String Name )
  {
  name = Name;
  attribute   = new StringBuffer();
  content     = new StringBuffer();
  morecontent = new StringBuffer(); //for text
  int deck_no = 0;
  }
}
Wrapper()
 {
  stck = new Stack();
  wmlrules = new Hashtable(); //initialize the table
  wmlrules.put("html","wml");
  wmlrules.put("body","card");
  wmlrules.put("b","b");
  wmlrules.put("h","b");
  wmlrules.put("h2","b");
  wmlrules.put("h3","b");
  wmlrules.put("h4","b");
  wmlrules.put("h5","b");
  wmlrules.put("h6","b");
  wmlrules.put("p","p");
  wmlrules.put("a","a");
  wmlrules.put("br","br");
  /*this table contains a list of keywords based on which
  the relevence of the content in determined. */
  dictionary = new Hashtable();
  dictionary.put("sports","1");
  dictionary.put("business","2");
  dictionary.put("stocks","3");
  dictionary.put("quotes","4");
  dictionary.put("weather","5");
  dictionary.put("world","6");
  dictionary.put("india","7");
  dictionary.put("entertainment","8");
  dictionary.put("arts","9");
  dictionary.put("news","10");
  dictionary.put("canvas","11");

 }
/* method that creates the wml deck
   stores it in an array the wml tags with their content
   this array is passed to method print() that writes in
   a file.
   */
void makewmlcard(Tags[] tagarray,int length) throws IOException
 {
  int body=0,flag=0,Len=0;
  Tags[] tagarray1= new Tags[length];
  int len=0;
  wmlarray = new wmltags[length+500];
  wmlarray1 = new wmltags[50];
  int Bufsize=0;
  for(int kk=0;kk<length;kk++)
  {
   //to remove video reports from the timesofindia page
   Len=tagarray[kk].morecontent.toString().length();
   if((tagarray[kk].content.toString().length()<5)&&(Len>0)&&(Len<34))
    System.out.println("tag "+tagarray[kk].morecontent.toString());
   else
```

```
agarray[kk].name.equals("a")&&tagarray[kk].content.toString().length()<5)
         System.out.println("tag1 "+tagarray[kk].content.toString());
      else
        tagarray1[len++]=tagarray[kk];
  }
  for(int k=0;k<len;k++)
  {
    if(tagarray1[k].name.equals("body")&&(body==0))
      {
        body++;
        String stag = (String)wmlrules.get(tagarray1[k].name);
        wmltags wmlcard = new
ags(stag,tagarray1[k].attribute,tagarray1[k].content,tagarray1[k].moreconten

        wmlcard.attribute.append(" id=\"main\" title=\"Main\" ");
        wmlarray[pointer++]= wmlcard;
        wmltags wmlcardp = new wmltags("p");
        wmlarray[pointer++]= wmlcardp;
      }

    else
      {
        String stag = (String)wmlrules.get(tagarray1[k].name);

        // remove bold tags with no enclosing anchor tags
        if(
rray1[k].name.equals("b"))&&(tagarray1[k].content.length()!=0)&&(!tagarray1
.name.equals("a")) )
        {
          System.out.println( "in here if
array1[k].content.toString());
          System.out.println(" in  "+tagarray1[k-1].name);
          stag="";
        }
        if(
rray1[k].name.equals("b"))&&(tagarray1[k].content.length()!=0)&&(tagarray1[
name.equals("a"))&&(tagarray1[k-1].content.length()!=0) )
        {
          stag="";
        }
        else
        {
          if( (!stag.equals(""))&&(!stag.equals("card")) )
            {
              if( (tagarray1[k].morecontent.length()==0)&&(flag>6) )
              {
               System.out.println("  end .."+wmlarray[pointer-
recontent.toString());
                break;
              }
              wmltags wmltag = new
gs(stag,tagarray1[k].attribute,tagarray1[k].content,tagarray1[k].moreconten

              if( (wmltag.morecontent.toString().length()>16)&&(flag==0))
              {
                // new card required for headlines
```

```
                    flag++;
                    System.out.println("card #1
"+wmltag.morecontent.toString());
                     start = pointer;
                     wmltags wmlcd1 = new wmltags("wml");
                     wmlarray[pointer++]=wmlcd1;
                    wmltags wmlcard = new wmltags("card");
                    wmlcard.attribute.append(" id=\" "+"cd"+CARD_NO++ +"\"")
                    wmltags wmlcd = new wmltags("p");
                    wmlcd.attribute.append(" mode=\"nowrap\"");
                    wmlarray[pointer++]= wmlcard;
                    wmlarray[pointer++]= wmlcd;
                  }
                  if( (flag>0)&&wmltag.name.equals("a"))
                  {
                    /*
                      make card for the morecontent. the card is
                      written to a new file(deck).
                      */
                    flag++;
                    wmltags wmlnewcard = new wmltags("card");
                    wmlnewcard.deck_no = DECK_NO;
                    StringBuffer href = new StringBuffer("href =
\"http://127.0.0.1:8080/"+u.baseaddr+DECK_NO++
+".wml/#cd"+CARD_NO+anchorcard+"\"");
                    wmlnewcard.attribute.append(" id=\""+"cd"+CARD_NO+anchor
+"\"");

                    wmltags wmlcd1 = new wmltags("p");
                    wmlcd1.morecontent=wmltag.morecontent;
                    wmltag.attribute=href;
                    wmlarray1[pointer1++]= wmlnewcard;
                    wmlarray1[pointer1++]= wmlcd1;
                    anchorcard++;
                  }
                  wmlarray[pointer++]=wmltag;
                }
            }
          }
      }
      System.out.println("lengeth "+wmlarray.length);
      wmltags wmltag = new wmltags("nil");
      wmlarray[pointer]=wmltag;
      wmltags wmlnewtag = new wmltags("nil");
      wmlarray1[pointer1]=wmltag;
      print(wmlarray);
      print(wmlarray1,pointer1);
      wmlfil.close();
      System.out.println("size of wml deck in bytes "+buffersize);
      removebold();
    } //end method
  /* method writes the array of tags in a file
   */
  void print(wmltags[] array) throws IOException
    {
      wmlfil = new FileOutputStream("title.wml");
      outwml = new PrintWriter(wmlfil,true);
      boolean state=true;
  String Header="<?xml version=\"1.0\"?>";
  String head = "<!DOCTYPE wml PUBLIC \"-//WAPFORUM//DTD WML 1.1//EN\"
.tp://www.wapforum.org/DTD/wml_1.1.xml\">";

  outwml.println(Header);
  outwml.println(head);
  for(int k=start;k<=pointer-1;k++)
  {
    state=true;
    stck.push(array[k].name);
    if((array[k].attribute.length()==0)) {

      if(
ay[k].name.equals("b"))&&(array[k].content.length()!=0)&&(!array[k-
.ame.equals("a")) )
        {
        System.out.println( "in here 0 "+stck.pop());
        }
      if(
ay[k].name.equals("b"))&&(array[k].content.length()==0)&&(array[k+1].name.eq
("a"))&&(array[k+1].content.length()==0 ))
        {
        System.out.println( "in here 1 "+stck.pop());
        }
      if(
ay[k].name.equals("b"))&&(array[k].content.length()==0)&&(!array[k+1].name.e
s("a")) )
        {
        //bold tags not followed by anchor tags to be ignored.
        System.out.println( "in here 2 "+stck.pop());
        }
      else

       outwml.println("<"+array[k].name+">");
    }
    else

rray[k].content.length()==0&&array[k].name.equals("a")&&(!array[k+1].name.eq
("b")) )
        {
        // anchor tags with no content should be ignored
        state=false;
        stck.pop();
        }
      else
         {

array[k].content.length()==0)&&array[k].name.equals("a")&&(array[k+1].name.e
s("b"))&&(array[k+1].content.length()==0) )
          {
            stck.pop();
            state=false;
          }
        else
          {
ray[k].attribute.toString()+">");
```

```
                outwml.println("<"+array[k].name+"
"+array[k].attribute.toString()+">");
              }
            buffersize+=array[k].attribute.length() + array[k].name.length();
            }

        if( (array[k].content.length()!=0))
          {
            outwml.println(array[k].content.toString());
            buffersize+=array[k].content.length();
            if(stck.peek().equals(array[k].name))
            {
              outwml.println("</"+ stck.pop()+">");
              if(stck.peek().equals("b"))
                outwml.println("</"+stck.pop()+">");
              if(stck.peek().equals("a"))
                outwml.println("</"+ stck.pop()+">");
              if(stck.peek().equals("p")&&array[k+1].name.equals("card"))
                outwml.println("</"+ stck.pop()+">");
            }

          }
        if(array[k+1].name.equals("card")&&(k>start))
          {
            while(!stck.peek().equals("card"))
            outwml.println("</"+ stck.pop()+">");
            outwml.println("</"+ stck.pop()+">");
          }
      }

      while(!stck.empty())
      outwml.println("</"+ stck.pop()+">");
      System.out.println("No. of cards  "+CARD_NO);

    }// end method

/* method to write morecontent in different files
 */
  void print(wmltags[] array,int len) throws IOException
    {
       String Header="<?xml version=\"1.0\"?>";
      String head = "<!DOCTYPE wml PUBLIC \"-//WAPFORUM//DTD WML 1.1//EN\"
\"http://www.wapforum.org/DTD/wml_1.1.xml\">";

      for(int ii=0;ii<len;ii++)
      {

        stck.push(array[ii].name);
        if(array[ii].name.equals("card"))
          {
            outwml = new PrintWriter(new
FileWriter("webpages\\"+u.baseaddr+array[ii].deck_no+".wml"),true);
            outwml.println(Header);
            outwml.println(head);
            outwml.println("<wml>");
          }
        if(array[ii].attribute.length()==0)
      outwml.println("<"+stck.peek()+">");
    else
      {
        outwml.println("<"+stck.peek()+" "+array[ii].attribute.toString()+
;
      }

    if(array[ii].morecontent.length()!=0)
      outwml.println(array[ii].morecontent.toString());
    if(array[ii+1].name.equals("card"))
      {
        while(!stck.empty())
        outwml.println("</"+stck.pop()+">");
        outwml.println("</wml>");
      }
    buffersize+=array[ii].attribute.length() +
y[ii].name.length()+array[ii].morecontent.length();
  }
  while(!stck.empty())
  outwml.println("</"+stck.pop()+">");
  outwml.println("</wml>");

}// end method
 to remove bold tags since the UP browser does not recognize
<b> tag
*/
id removebold() throws IOException,FileNotFoundException
{
  String lineread=";",strg="y";
  outwml = new PrintWriter(new FileOutputStream(u.baseaddr+".wml"),true);
  FileInputStream infile = new FileInputStream("title.wml");
  BufferedReader fread = new BufferedReader( new InputStreamReader(infile));
  lineread=fread.readLine();
  while(!lineread.equals("</wml>"))
  {
    if(lineread.equals("<b>")||(lineread.equals("</b>")))
      outwml.print(" ");
    else
      outwml.println(lineread);
    //}
    lineread=fread.readLine();

  }
 outwml.println("</wml>");
// end method

blic void get(wmltags[] arry)throws MalformedURLException,IOException

 String keyword = "1",stg="_";
 System.out.println("in ...................."+start);

 for(int ii=0;ii<start;ii++)
 {
   Enumeration e = dictionary.keys();
   keyword = (arry[ii].content.toString()).toLowerCase();

   if(keyword.length()>3)
```

```
          {
            keyword = keyword.substring(1,keyword.length());
            if(dictionary.containsKey(keyword))
            {

                  System.out.println("keyword in arry :"+keyword);
                  System.out.println("keyword
:"+(String)dictionary.get(keyword));
                  System.out.println(arry[ii].attribute.toString());
                  getPage(arry[ii].attribute.toString());


            }
          }
      }

    }//end method
  InputStream getPage(String addr)throws MalformedURLException,IOException
    {
      URL ul;
      String str = ".";
      if(addr.startsWith("http",6))
      ul = new URL(addr.substring(6,addr.length()));
      else
      {
        str = u.baseaddr+addr.substring(6,addr.length());
        System.out.println("str "+str);
        ul = new URL("http://"+str);
      }
      return ul.openStream();

    }//end method

  public boolean equal(String str1,String str2)
    {
      System.out.println(str1.length()+" "+str2.length());
      if(str1.length()!=str2.length())
       return false;
      else
      {
      System.out.println("reached");
      for(int i=0;i<str2.length();i++)
        {
          System.out.println(str2.charAt(i));
          if(str1.charAt(i)!=str2.charAt(i))
            return false;
        }
      return true;
      }

    }// end method

} //******************end class Wml ****************************


}// **************end class test *******
```