Neo Browser: Java Based Web Browser for Mobile Phones

Shailendra Singh, Prashant Bhardwaj, and Mohammed Abdul Qadeer

***Abstract*—This paper presents a design and development of a web browser for J2ME enabled devices especially mobile phones. The paper gives a basic overview of the current stage in web browsing and the progress achieved and future advancements in our project we are looking forward to. At present, mobile phones include inbuilt web browsers but with limited functionalities, low support for multimedia and tabbed browsing. Our current work has completed web page parsing, audio and video streaming. The tabbing functionality is currently in development phase but the current status promises positive achievements in future.**

***Index Terms*—Audio and video streaming on mobile devices, J2ME web browser, mobile web browser, tabbed browsing for mobile phones.**

I. Introduction

At present, nearly all networking capable mobile devices have their own web browsers. But such browsers have limited functionality due less hardware resources and also due to the limited bandwidth provided by the networks. In this paper, we present the need of a web browser that can be ported on any platform and which includes advanced functionalities like multimedia streaming, tabbed browsing etc. along with the basic web browsing functionalities.

II. Technologies Used

A. Java 2 Micro Edition

J2ME is the Java technology for mobile devices. It is organized in configurations and profiles. A configuration consists of Java Runtime Environment and core classes that operate on each device. A profile consists of classes that enable developers to implement features found on a related group of small computing devices. J2ME is organized in two configurations namely Connected Device Configuration (CDC, utilizes higher resources) and Connected Limited Device Configuration (CLDC, for resource constrained devices). Also, seven J2ME profiles have been defined at the time of this writing. They are the *Foundation Profile*[1] (basic CDC profile), *Game Profile*[1] (CDC profile for game applications), *Mobile Information Device Profile*[1] (MIDP, described below), *PDA Profile*[1] (CLDC profile for sophisticated devices), *Personal Profile*[1] (CDC profile for complex user interfaces), *Personal Basis Profile[1]*(CDC profile for simple user interface), and the *RMI Profile* [1] (CDC profile for Remote Method Invocation). The basic J2ME functionalities utilized are:

- Connected Limited Device Configuration
- Mobile Information Device Profile
- Mobile Media API

The first two technologies are used to design the Graphical User Interface whereas the second technology is used for the audio and video streaming.

B. GPRS Network

General Packet Radio Services (GPRS) is a packet-based wireless communication service based on Global System for Mobile (GSM) that provides data rates from 56 up to 114 kbps and continuous connection to the internet for mobile phones.

We have used the GPRS network as it is the most common network and its low bandwidth makes its ideal to test the application. Our browser is targeted to work on the common low data rate GPRS network. Although, no network-specific customizations have been added, the speed enhancement analysis is underway to suit the needs of a busy and low data rate network. This also forms the main reason for targeting the GPRS network.

C. Apache Tomcat Server

It is a local server utility from Apache. The tomcat server installed on the testing machine behaves the same way as a general Internet/LAN server. It forms a good choice of network based testing in the initial phase of development, for quick testing.

The version of Tomcat we have used to test our application on is Apache Tomcat 6.0.26.

D. Light Weight User Interface Toolkit

The Light Weight User Interface Toolkit (LWUIT) [2] is a versatile and compact API for creating attractive application user interfaces for mobile devices. LWUIT provides sophisticated Swing-like capabilities without the tremendous power and complexity of Swing. Designed from the ground up as an efficient mobile user interface toolkit, LWUIT provides many useful Swing-like features. LWUIT offers a basic set of components, flexible layouts, style and theme, animated screen transitions, and a simple and useful event-handling mechanism. The toolkit has been tested and debugged on a variety of mobile devices. LWUIT is very different from Swing and has taken on features unavailable in Swing such as theme, painters, animations etc. However features such as MVC, layout managers, renders and the EDT are directly related to Swing. LWUIT is based on a Component/Container hierarchy composite architecture. Containers are Components and can be nested to create elaborate layouts. Components can be styled both via external styles/themes and programmatically by developers.

Manuscript received June 26, 2012; revised September 27, 2012.

The authors are with the Department of Computer Engineering, Aligarh Muslim University, India (e-mail: shailendrasingh@zhcet.ac.in, prashantbhardwaj@zhcet.ac.in; maqadeer@gmail.com).

III. Progress Achieved with Technical Details

Our project is still in the developing phase, hence the intended project goals have not been achieved yet. Although, a major part has been covered in its raw form using Component Based Development, i.e. most of the functionalities have been separately designed. The part of the project that we have successfully covered have been tested both on mobile phone (Nokia 5800 XpressMusic) as well as on the DefaultColorPhone of the Sun Java Wireless Toolkit 2.5 using the NetBeans IDE.

Some of the progress that has been achieved includes:

A. Web Page Visiting and Parsing

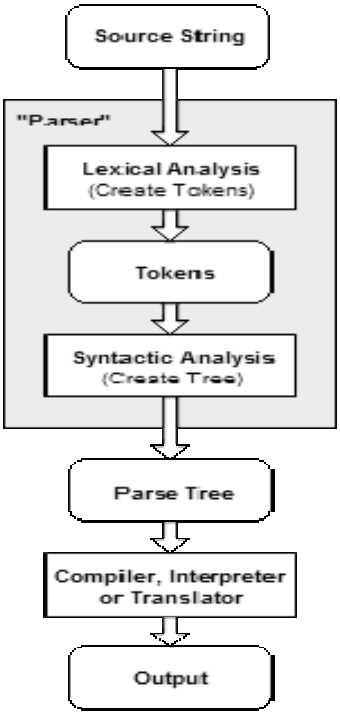

Fig. 1. showing the working of a parser.

The http Connection is used to download the content from a local or web server. As the web pages are in a string format so the downloaded web pages are parsed to display the content present on the web page.

We have studied the Pico Browser [3] and the PL Browser [4] and then developed a parser for parsing the xHTML that is extensively used for designing the web sites for mobile.

Visiting a website: one can visit a website (mobile web site) by specifying the URL of the page.

Link Redirection: Link redirection from one page to other page is also possible.

History: Suppose one has visited a page and moved to another page then he can visit the page again without specifying the URL again by just clicking the back button. Similarly one can also visit the page next page.

B. Audio and Video Streaming

Streaming is a technique of transferring data so that it can be processed as a steady and continuous stream. Streaming technologies are gaining much importance with the growth of the internet because most of the users do not have desired fast access to download large multimedia files quickly. With streaming, the client browser can start displaying the data before the entire file has been downloaded.

Fig. 2. The figure given above shows the graphical user interface of the web browser and the entry of the URL for Google mobile. The GUI includes the various buttons for visiting previous URL, cancel the loading of page, next URL, refresh button, home page button and text field for entry of a URL.

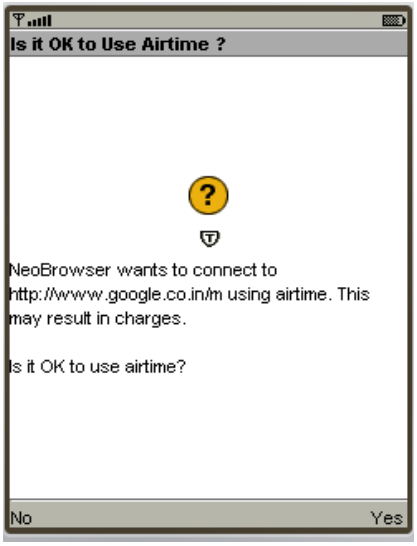

Fig. 3. The figure given above shows the response of the web browser when the URL has been entered. The figure shows that the web browser is querying for an internet connection.

Fig. 4. This figure shows the Google page being fetched by the web browser after granting the request to connect to the internet.

For streaming to become practical, the client side i.e. the web browser which is receiving the data must be able to collect the data and send it to the application as a steady stream that is processing the data and converting it to audio or video. Hence if the streaming client receives the data more quickly than required then it needs to save the excess data in a buffer. If the data doesn't come quickly enough, however, the presentation of the data will not be smooth.

J2ME does not offer direct support for multimedia streaming, but it is protocol independent thus offering support for Real Time Protocol (RTP) also. But, due to high resource requirements and platform support required by RTP, we use a different download and play approach for multimedia streaming [5] over HTTP.

This functionality has been achieved by downloading the media files to a temporary location and then periodically querying the browser's media control whether it recognizes the downloaded content. One of the thread downloads the file while the other thread plays the downloaded file.

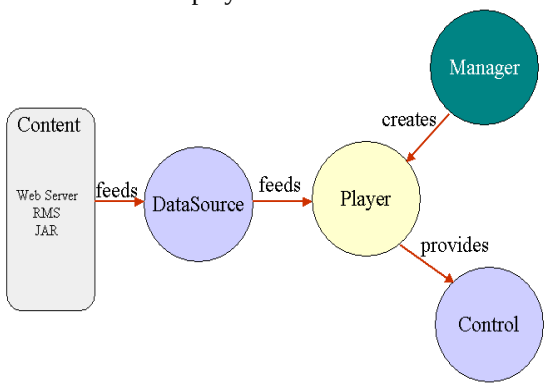

Fig. 5. This figure is showing MMAPI architecture for Multimedia Streaming

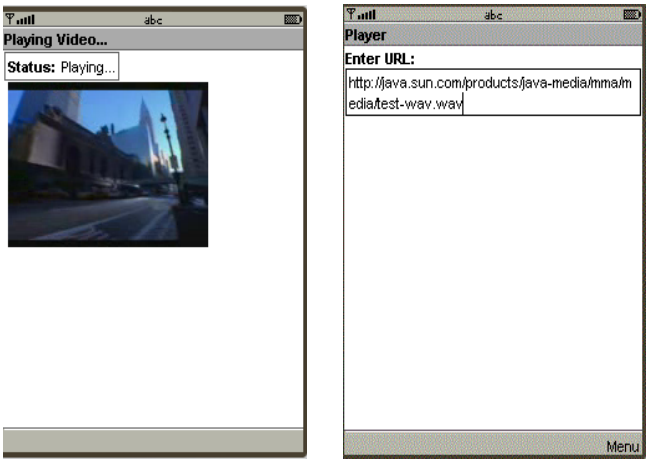

Fig. 6. The figures above show audio and video streaming being performed. The figure on the left shows the video being played located the link http://java.sun.com/products/java-media/mma/media/test-mpeg.mpg and the figure on the right shows the audio being played located at the link http://java.sun.com/products/java-media/mma/media/test-wav.wav.

C. Tabbed Browsing

Tabbed browsing is a function of a web browser that allows the user to surf and view multiple pages by loading the web sites into "tabbed" sections of one page, rather than multiple pages. *Tabbed browsing* allows the user to quickly switch between several open web pages within a web browser window by clicking on tabs. Tabs appear in the browser window when more than one webpage is opened. Each open webpage has its own tab, with the active tab displaying its page, the inactive tabs hidden behind. By clicking on a tab the user switches that page up front for quick viewing.

Tabbed browsing is extremely useful if one is reading a webpage and would like to click a link without losing one's place, one can open the link in a *tabbed* window. After reading the link one can close the tab or just click on the original tab to switch back to the previous page, keeping both pages available. *Tabbed browsing* eliminates the need to repeatedly reload pages.

The need of tabbed browsing specifically arises in the arena of the mobile web browsing because generally the web browsers present today have only a single instance of the web browser window and hence to view various pages at the same time tabbed browsing is needed.

We have worked on providing the different tabs to the user. The tabs are in the form of different screens which can be switched from one another using the cursor keys on an emulator and the navigation keys on the mobile phone.
When the user press a key the class containing the graphical user interface is called which opens in a new thread. Thus having a tabbed representation [6].

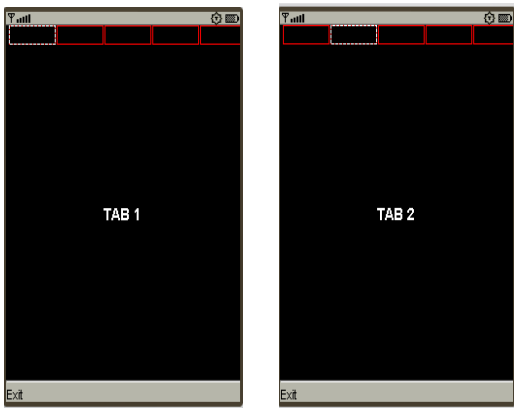

Fig. 7. The figures shown above gives a representation of tabbed browsing, where the white boxes show the active tab and the red boxes show the inactive opened tabs. The works space is showing the tab names being written which can be used to show the contents of the web page that is accessed in the respective tabs and the empty space in the boxes can be utilized for specifying the name or address of the web pages being displayed by the respective tab. Since the mobile devices have limited RAM we have tried to implement only a limited number of tabs.

IV. Remaining Work

A. Tabbed Browsing

Although we have provided different tabs but successful networking on these tabs is still to be done.

This can be achieved by storing the presently opened tab and using the concept of threading so that they can be downloaded in background and when opened the tab it can be parsed [7].

B. Integrating the Components

We have worked out all the functionalities in different modules. We have created separate reusable components for specific tasks that will be integrated at the end. All the above mentioned functionalities have been tested separately but their integration inside one common package is yet to be done, that we will consider as the final stage browser. Although, component integration is generally not an issue for well

written code, synchronization issues may still arise.

V. Future Prospects

We look forward towards implementing the following features and utilities in future;

A. In-built Download Manager

A download manager is a computer program dedicated to the task of downloading possibly unrelated stand-alone files from the Internet for storage. The typical download manager at a minimum provides means to recover from errors without losing the work already completed, and can optionally split the file to be downloaded into 2 or more segments, which are then moved in parallel, potentially making the process faster within the limits of the available bandwidth.

Our download managers will include the features like pausing the downloading of files, resuming the paused downloads, downloading several files simultaneously and scheduled downloads.

B. Mobile Instant Messaging

In a mobile environment, the user is constrained by bandwidth and the UI. However, the user has the advantage of mobility. Mobile Instant Messaging users may desire to message with other Mobile Instant Messaging users as well as fixed Instant Messaging users on networks such as AOL, Microsoft and Yahoo. Mobile Instant Messaging allows users to address messages to other users using an alias (or user name) and address book, allowing the sender to know when his/her friends are available. This is an important feature, so we will make this as the part of our web browser so that the user may not have to search for the addition softwares.

C. Parser Utility

At present, we have used xHTML parser and displaying xHTML content of mobile websites. We look forward to design a parser that will support building HTML, CSS and other formats of the web pages so as to display them on phone.

VI. Conclusion

The J2ME platform provides great networking capabilities that can be exploited on the present mobile phones. The lack of web browsers with advanced features for mobile platforms addresses the need of more development in the field.

Multimedia Streaming on mobile devices is limited to high speed internet connections and not on developed GPRS networks. Although, the capabilities of such networks cannot be enhanced just by software, efficient methods are needed to send large files over the communication channel.

Our project has fulfilled the basic task of web browsing, multimedia downloading and streaming. But yet the Tabbed browsing feature is yet to be developed fully.

Acknowledgment

We would like to express our deepest gratitude and warmest appreciation to every person who has contributed in any way and inspired us to the overall success of the undertaken project.

We are highly obliged to our supervision for providing his expert guidance and motivation to do this project.

We are thankful to our friends, who have been unselfishly extending their efforts and understanding, to our parents who have always been very understanding and supportive both emotionally and financially.

And above all, to the Almighty God, who never cease in loving us and for the continued guidance and protection.

References

[1] J. Keogh, "J2ME: The complete reference," *McGraw Hill/Osborne*, 2003.
[2] Sun Developer Community. [Online]. Available: http://developers.sun.com
[3] Pico Browser. [Online]. Available: http://www.beartronics.com/bearlib.sourceforge.net
[4] PL Browser. [Online]. Available: http://www/pico.html
[5] A. Pachuri, P. Varshney, and M. A. Qadeer, "Mobile web browser with streaming capability," in *Proc. ICWET 2010, ACM*
[6] Nokia Forum. [Online]. Available: http://wiki.forum.nokia.com
[7] Borland Software Corporation, "Developing mobile application," *Borland JBuilder: Version 9.*

Shailendra Singh is expected to graduate with a Bachelor of Technology in Computer Engineering in July 2012 from Department of Computer Engineering, Aligarh Muslim University, India. He has a good academic background and is a member of the Research and Development Cell at Faculty of Engineering and Technology, Aligarh Muslim University. He is responsible for the maintaining the official website of the college. He takes care of the mailserver of the college. He is also responsible for the MIT-OCW course ware at the college and maintaining the ERP solution for the college. As a part of academic career he has developed a compiler for Python which is based on the Low Level Virtual Machine infrastructure. Soon he is going to serve Tata Consultancy Services as an Assistant System Engineer.

Prashant Bhardwaj is expected to graduate with a Bachelor of Technology in Computer Engineering in July 2012 from Department of Computer Engineering, Aligarh Muslim University, India. He has a good academic background as is evident from his All India Rank 13 in Graduate Aptitude Test in Engineering which was conducted by Indian Institute of Technology, Delhi, a premier institute in the country. As a part of academic career he has developed a compiler for Python which is based on the Low Level Virtual Machine infrastructure. Soon he is going to be a student of Indian Institute of Science, Banglore for his MS Degree in Machine Learning.

Mohammed A. Qadeer is an Asst. Professor with the Department of Computer Engineering, Aligarh Muslim University, India. Earlier, he was working with Cisco Systems Inc. as a Network Consulting Engineer with the Advanced Services division in the APAC region. He received his B.Sc. Engineering (Computer Engineering) from Aligarh Muslim University in 1996. He has an experience of 15~ years in the area of computer networks and systems. He served as a Technical Co-Chair for IEEE WOCN 2012, 2011, 2010, Technical Co-Chair IEEE AH-ICI 2012, 2011, International Steering Committee for ICACT 2012, 2011, 2010 and as TPC member for CCNC 2011, 2010, INMIC 2009, AH-ICI 2009, WIA 2009 and MMA 2009.

His areas of research are computer networks, wireless networks, mobile computing, next generation networks, IMS, LTE, WiMAX, 4G, WiBro etc.