WAP for Ethiopic Content

An application for Accessing Marketing Information

By

Teferi Assefa

A PROJECT REPORT

SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES OF ADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF SCIENCE IN COMPUTER SCIENCE

2005

Addis Ababa University

School of Graduate studies

Addis Ababa University

School of Graduate Studies

Department of Computer Science

WAP for Ethiopic Content: An Application for Accessing Marketing Information

By

Teferi Assefa

Name and Signature of Members of the Examining Board

1. Dr. Solomon Atnafu, Advisor _____
2. Dr. _____

Acknowledgment

First of all my appreciation and thanks is for my advisor Dr. Solomon Atnafu, for his kindly and thorough help in each step of this work. Without him this work may not come to the end. I also extend my acknowledgment and special thanks to my wife W/ro Emnet Mulugeta, she undoubtedly assisting me in writing the report and by supporting with the necessary moral and material help.

I extend my deep appreciation to Ato Tamrat Abebe, for his help in editing the document thoroughly. Finally I will pass my appreciation to all of my colleagues who have guided me through contribution of different things.

Table of Contents

Title **Page**

List of Tables

Page

List of Figures

Page

Abstract

Accessing current and accurate information being any where at anytime is becoming a growing interest nowadays. WAP (Wireless Application Protocol) is an application protocol that creates an opportunity to access information of an interest from WAP servers using mobile phones. WAP supports a UTF-8 encoding, which can be used for developing a multilingual WAP pages.

In this work, the necessary standards, tools and configuration requirements for Ethiopic WAP content development are identified and studied. As a result of this study we observed that WAP technology could be applicable for information delivery in Ethiopic content on mobile phones.

To demonstrate the result of this work, we have developed a WAP application for agricultural market information delivery in Amharic language.

Key words: WAP for Ethiopic, WAP based Marketing Information, WAP localization, WAPE, XHTML MP for Ethiopic

1. Introduction

Information is one of the key factors to facilitate the economic growth of a country. Information is a base for performing different activities in several disciplines. For instance, information regarding the demand of a particular good is essential for a producer to decide when and where to sell the product. As a result of which the producer would be able to maximize profit and leads towards a better economic growth of a country.

Internet is one way of retrieving information from different areas through the web. Most of the technology developed for the Internet has been designed for desktop and larger computers supporting medium to high bandwidth connectivity on reliable data networks. Recently, technologies that enable handheld wireless devices to retrieve information are underway. However, these handheld wireless devices present a more constrained computing environment compared to desktop computers. In addition, providing Internet and WWW (World Wide Web) services on a wireless data network presents many challenges.

These days, the demand for the wireless data link is hotter than ever. As we know wireless devices give opportunities to remain connected as we move from place to place. These desires of individuals are met by mobile terminals and internet portals with mobile content. The retrieval of information through Internet from anywhere and at any time is an interest of every body [22], because getting accurate information from anywhere on time is an important element for making a better decision.

WAP (Wireless Application Protocol) creates new business opportunities for corporations by providing additional channel for the existing services. The possibility of this additional service is that it can reach customers 24 hours a day wherever they are. Since WAP is an open protocol for wireless messaging, it provides the same technology to all vendors regardless of the network system. This common standard offers a better market scale that encourages manufacturers, application developers and content providers to invest in developing WAP compatible products.

At the present time, accessing web content from anywhere and at any time using mobile devices is possible using WAP technology. It is thus possible to reliably and efficiently communicate data over wireless WANs (Wide Area Networks). Retrieval of information using local language is also another critical issue for facilitating different tasks for those who work in their local languages. Thanks to the Unicode and other standards, developing multilingual WAP services and getting WAP enabled mobile terminals that support multilingual character set is no more a problem.

Delivery of information using local language increases the audience and the user of this technology, which is an issue raised by many users. But today, delivering multilingual information on different mobile devices from the Internet is possible using WAP services. So, retrieving Ethiopic content from the WAP servers using mobile terminals from anywhere and any time is an interest of Ethiopic content users for different applications.

1.1. Statement of the Problem

While mobile phones are becoming dominant means of accessing information using WAP applications, based on WAP technology no work is done so far for Ethiopic content delivery.

WAP content delivery using local languages can be used for many applications, such as market information delivery. However, no work is done in this regard for delivery of information using an Ethiopic content WAP application.

1.2. Objectives of the Project

The general objective of this project is to explore the capability of WAP technology in accessing Ethiopic content information from the web using mobile devices.

The specific objectives are to:

- Investigate the possibility of developing WAP pages for Ethiopic content
- Explore the features of WAP service for information delivery.
- Analyse the requirements of marketing information delivery for cooperative societies under FCC.
- Design an appropriate method for delivering marketing information
- Develop a prototype to demonstrate the possible use of WAP in Ethiopic content for marketing information delivery

1.3. Application of the Study

Obviously, services that you use while being “mobile” are in general suited for the mobile Internet access than accessing on fixed devices. Some great applications are starting to emerge, from which the mobile Internet users will benefit [5, 6, and 18].

Accessing information from anywhere and at anytime from the Internet is currently a practical event. By combining the best of existing technologies with the incredible potential of the developer community, WAP gives subscribers an efficient and exciting new method of accessing information from anywhere at any time. Services can be available from any location, and users can have wide opportunities when it comes to staying in contact with information. Providing wireless Internet service for Mobile terminals, and enabling them to browse an Ethiopic content WAP pages, increases the utilization of the service by most Ethiopians who know only their native language.

Using WAP service, it is possible to develop an application that can be used to access current information, such as marketing information, weather condition, and stock price of different items and so on. Different users from different environments may be interested to access information or to get different types of services using their mobile devices.

In addition, a company may use such service for delivery of advertisement using both local and international languages for its customers. This increases the number of audience of the advertisement in three ways.

1. These days most mobile devices are WAP enabled, and that mobile devices are affordable by many of the citizens.

2. The availability of the information in local language

3. The mobile network infrastructure is increasing with high speed in the country and covering the nation with respect to other available network.

In flight scheduling for example, due to different reasons the original flight schedule may be canceled. The agent or the airline ticket officer should rebook the passengers on alternate flights and can inform using the WAP services by sending necessary information to their mobile phones. In addition to this friends can exchange short message notification of different events using WAP services.

2. Literature Review

2.1. WAP Technology

The wireless communication and the Internet are the rapidly growing industries that are gaining more and more customers every day. The WAP intention is to combine these two markets and met the new demands in the field. This and other reasons initiate some of the largest vendors to unite and create the WAP Forum, the standardizing organization of the WAP [1, 6].

WAP specifies an application framework and network protocols for wireless devices such as mobile phones, pagers and PDA (Personal Digital Assistants). WAP's specifications extend existing mobile networking technologies and some Internet technologies such as XML (eXtensible Markup Language) and scripting content formats.

The WAP platform is an open specification that addresses wireless network characteristics by adapting existing network technologies (and introducing new ones where appropriate) to the special requirements of handheld wireless devices [6]. Therefore, WAP intends to standardize the way wireless devices (mobile phones, PDA, and so forth) access Internet data and services. WAP's reuse of existing Internet protocols will ease the development of WAP services for Java and other Web developers.

Facilitating the delivery of Internet data to wireless devices will certainly lead to the introduction of new technologies. Wireless handheld devices preset a more constrained computing environment and platforms, compared to desktop computers which most of the Internet technology was developed for. The handheld devices tend to have less powerful CPU's, less memory, very restricted power consumption, smaller and variant displays, phone keypads etc [20]. Furthermore, the wireless networks present additional constraints as communication infrastructures. They have less bandwidth, more latency and less connection stability and unpredictable availability. WAP intends to overcome these difficulties by being interoperable, have scaleable quality of service, efficient in the mobile network resources, reliable and secure [19].

WAP allows carriers to strengthen their service offerings by providing subscribers with the information they want and need while on the move. Infrastructure vendors will deliver the supporting network equipment. Application developers and content providers delivering the value added services are contributing to the WAP specification. Enabling information access from handheld devices requires a deep understanding of both technical and market issues that are unique to the wireless environment. The WAP specification was developed by the industry's best minds to address these issues [29].

Nowadays web pages are browsed on mobile terminals using the WAP. Because, WAP is a standardized way for delivering Internet data over wireless networks and capable of addressing the unique characteristics of mobile terminals and wireless networks [1,6].

2.1.1. The WAP Architecture

The WAP standard defines two essential elements: an end-to-end application protocol and an application environment based on a browser. The application protocol is a communication protocol stack that is embedded in each WAP-enabled wireless device (also known as the user agent). The server side implements the other end of the protocol, which is capable of communicating with any WAP client. The server side is known as a WAP gateway and routes requests from the client to an HTTP (Hyper Text Transfer Protocol) (or Web) server. The WAP gateway can be located either in an Operator premises (Figure 2.1) or in WAP application provider premises with the web server (Figure 2.2).

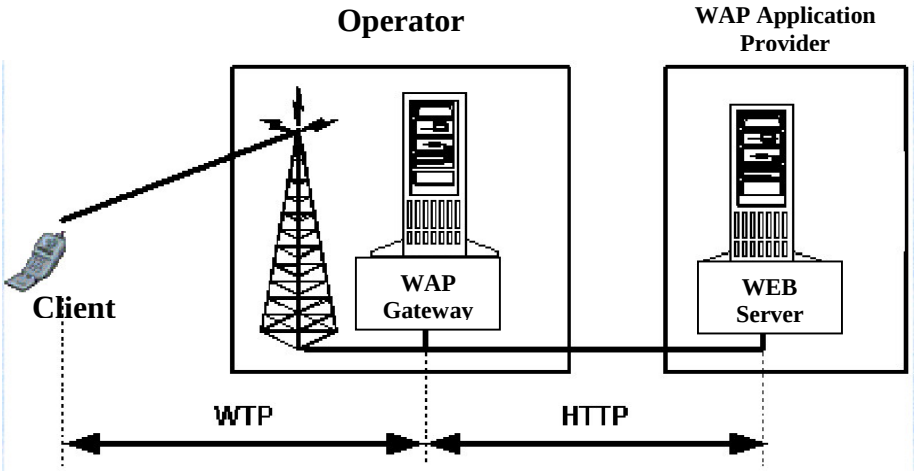

Figure 2.1 Gateway equipment in the operator premises

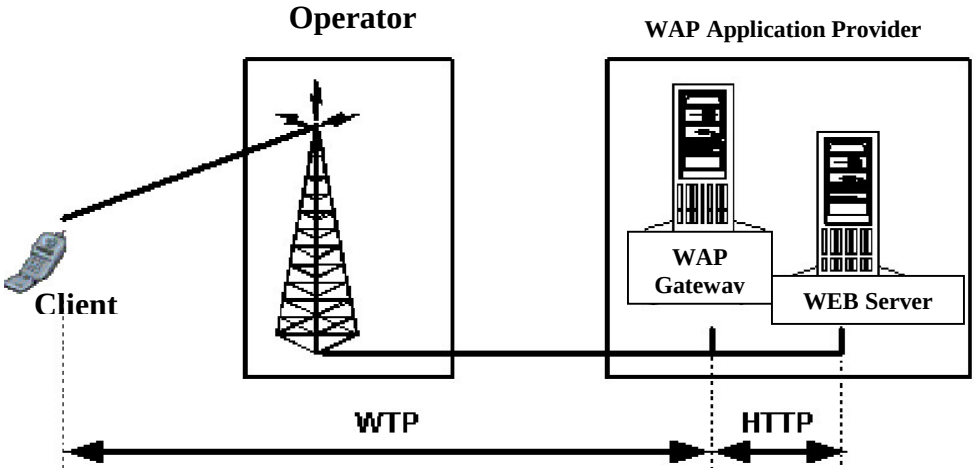

Figure 2.2 WAP Gateway in the WAP application provider premises

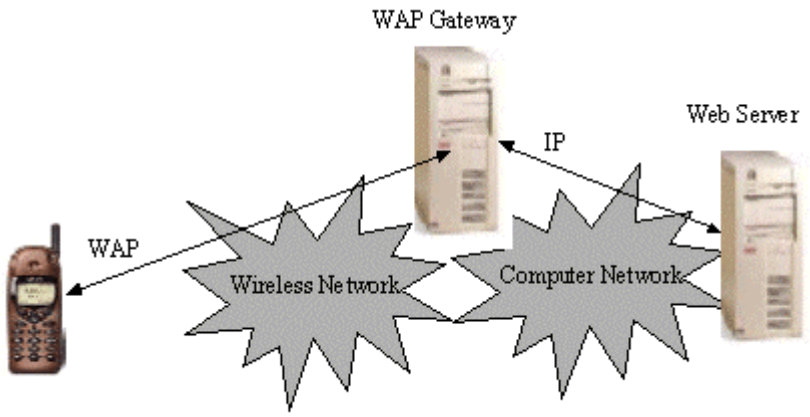

Figure 2.3 The WAP network structure

Figure 2.3 illustrates an example structure of a WAP network [19]. In the WAP network the client communicates with the WAP gateway in the wireless network. The WAP gateway translates WAP requests to WWW requests, so the WAP client is able to submit requests to the Web server. Also, the WAP gateway translates Web responses into WAP responses or a format understood by the WAP client.

The wireless application environment provides WAP micro browser for interaction between WAP (web applications) and wireless devices [5]. This browser relies on WAP markup languages such as WML (Wireless Markup Language), WML Script and XHTML MP (Extensible Hypertext Markup Language Mobile Profile) [9].

2.1.2. The WAP Programming Model

The WAP programming model is similar to the Web programming model with matching extensions, but it accommodates the characteristics of the wireless environment. Figure 2.4 illustrates this model.

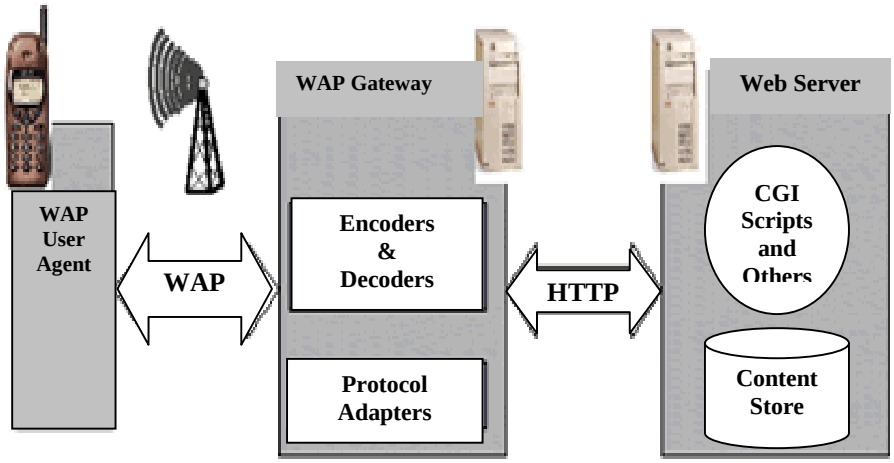

Figure 2.4 The WAP programming model [19]

The WAP programming model is based heavily on the Web programming model. But how does the WAP gateway work with HTML (Hyper Text Markup Language)? In some cases, the data services or content located on the Web server is HTML-based. Some WAP gateways could be made to convert HTML pages into a format that can be displayed on wireless devices. Because HTML wasn't really designed for small screens, the WAP protocol defines its own markup language [19].

WML, WML Script, and XHTML MP are the languages that are specifically designed to develop WAP applications for Mobile devices. These languages adhere to the XML standard and are designed to enable powerful applications within the constraints of handheld devices [13]. In most cases, the actual application or other content located on the Web server will be native WAP contents created with WML (XHTML MP) or

generated dynamically using WML Script, Java Servlets or JSP (Java Server Page), or other server side programming languages.

WML is an XML-based markup language that was designed especially to present WAP content on a wireless terminal. WML can preserve the content of variables between different WML pages. The basic unit of WML is the card that specifies a single interaction between the user and the user agent. Multiple cards are grouped together in decks, which is the top most element of a WML file. When the user agent receives a deck, it activates only the first card in the deck. There are no functions to check the validity of user input or to generate messages and dialog boxes locally in using WML. Therefore, to overcome this limitation, WML Script was developed [14].

WML Script, which is based on ECMA Script (the standard for java script), is a language that can be used to provide programmed functionality to WAP applications. It was defined to enable the execution of scripts on WAP devices. The goal of using WML Script is to reduce the number of turn around between the client and the server. It is part of the WAP specification, and it can be used to add script support to the client. Its difference from ECMA Script is that it is compiled into byte code before it is sent to the client. The main reason for this is to cope up with the narrowband communication channels and to keep client memory requirements to a minimum [13, 14].

XHTML is a markup language used to create richer web content on an ever-increasing range of platforms including mobile handsets [22]. It is similar with HTML in its tag definition and syntax, but it adds modularity and enforces strict adherence to language rules. It brings a clear structure to web pages, which is especially important for the small screens and limited power of mobile devices. The XHTML MP is a mobile adaptation of XHTML by excluding those features not appropriate for devices with small screens. It is a strict subset of XHTML that includes additional elements and attributes that are useful in mobile browsers with additional presentation elements and support for internal style sheets [13, 21].

Mobile browsing technology is evolving from WAP 1.x to WAP 2.0, by introducing different enhancements for mobile content development. Especially WAP 2.0 provides

support for protocols such as IP, TCP and HTTP [22]. This provides interoperable optimizations suitable to the wireless environment and to the environment that permits wireless devices to utilize existing Internet technologies. WAP 2.0 also provides different application environment, which enables delivery of information and interactive services to wireless devices [1].

WAP standard defines the future of wireless browsing technology based on the WML, XHTML MP and WAP CSS (WAP Cascading Style Sheet). Both WML and XHTML MP are a reformulation of the XML. XML is a language for marking up structures in text documents and supports the UTF-8 (8 bit Unicode Transformation Format) coding standard. The UTF-8 coding standard supports several languages character set including Ethiopic. So WML and XHTML MP can be used to create WAP pages that are encoded as UTF-8. Browsing from wireless terminals supporting UTF-8 encoding becomes possible [4].

2.1.3. The WAP Communication Model

WAP contents usually reside on WWW servers on the Internet. The WAP gateway placed between the mobile network and the web servers. It receives WAP requests using the binary WAP communication protocols, and translates the requests to the text based WWW protocols and forwards to the content servers using the TCP/IP network protocol.

The WAP gateway also waits for the WWW text protocol reply from the content server, and receives using the TCP/IP protocol. It formats the message to binary WAP protocol, and then sends the reformatted response to the WAP client via WDP (WAP Datagram Protocol), which is almost equivalent to UDP (User Datagram Protocol) of the Internet. WDP makes no attempt to confirm delivery, resend lost packets, or correct errors in transmission like the UDP [18].

The WAP protocols are designed to operate over a variety of different bearer services [14], including short messages, circuit-switched data and packet data. Each bearer offers different levels of quality of service with respect to throughput, error rate and delays. The WAP protocols are designed to compensate for or tolerate these varying levels of service.

The WDP specification lists the bearers that are supported and techniques used to allow WAP protocols to run over each bearer [7, 18].

2.2. WAP and Multilingual Application Development

By design, the WWW crosses international boundaries and supports multiple languages. To make your site or intranet application available to the largest audience, your pages need to be localized and internationalized. L10N (Localization) is the process of presenting content such as text and graphics using the local languages' character set while accounting for the cultural differences of the target audience.I18N (Internationalization) is the process of designing and implementing an application or content (including text and non-text elements) so that it can accommodate any local languages [17].

To provide support for I18N and L10N, the character sets must be encoded using UTF-8. The WAP specification version 1.2 and above, supports I18N, i.e. UTF-8 encoding. UTF-8 can be used for developing a multilingual web (WAP) pages using appropriate markup languages [17].

For example using the Openwave mobile browser, it is possible to develop, test and deploy WML applications that are localized to both for European and Asian languages and character sets. In addition with the Openwave mobile access Gateway and using the existing web standards, it is possible to handle language other than English [3].

Thus it has been possible to develop WAP pages using WML for simplified Chinese, traditional Chinese and Latin characters, using any editor that can save the file encoded as UTF-8. The encoding type included in the XML encoding header that defines the character set the file is using. Hence, a WAP enabled mobile terminal can display the content in the correct way [4].

2.3. WAP Services

WAP provides two types of services, which are pull and push (Figure 2.5). In pull service, the user can request and fetch for the WAP site, for the information that he needs to browse. In the Push service, origin server or PI (Push Initiator) initiates the connection and sends the push messages to the mobile device. The push service can be, for example, messaging, stock price and traffic update alerts [1].

As shown in Figure 2.5, the client requests a server for a response in a pull service, and a server initiates a connection with a client for a push service.

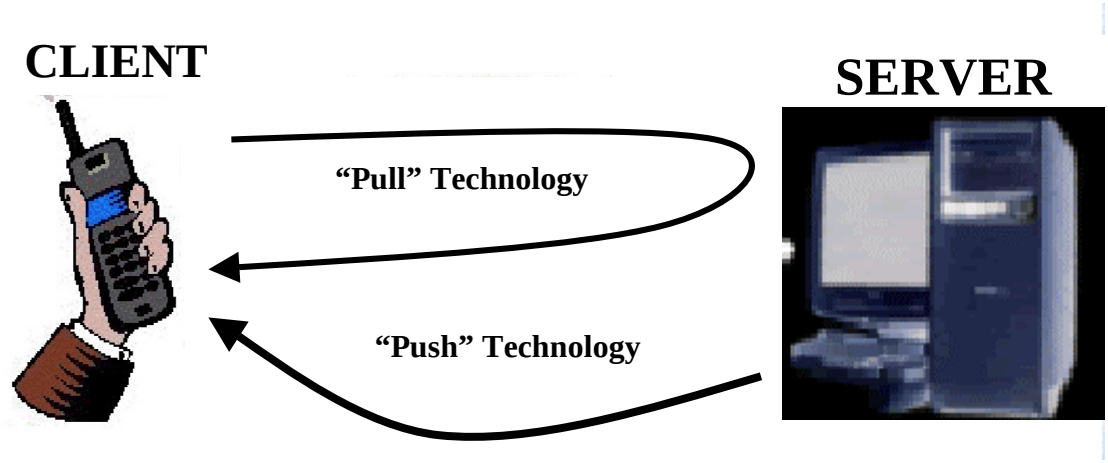

Figure 2.5 The structure of a WAP pull and push service

2.3.1. Pull Service

WAP pull is a traditional WAP service. It works similar to the normal client/server model. In the normal client/server model, a client requests a service or information from a server, which then the server responds by transmitting information to the client. This is known as “pull” technology. That is, the client pulls information from the server. The WWW is the best example of pull technology, where a user enters a URL (Uniform Resource Locator) (the request), which is sent to a server, and the server answers by sending a web page (the response) to the user. Similarly, in WAP pull, the user agent requests content from a WAP server. Then the WAP server reply through the WAP proxy to the user agent and the user agent can access the content.

When developing an application, which has connection with a database server, a pull service of WAP is essential. For example banking (transfer money between accounts...), finance (buy and sell stocks, exchange rates…), shopping (buy books, searching for records…), and Ticketing (cinema tickets, concert tickets…) are some of the applications, which can be developed using the pull technology of WAP [12].

2.3.2. Push Service

WAP push, available since WAP 1.2, allows WAP content to be pushed to the mobile handset with minimum user intervention. A WAP push is basically a specially encoded message that includes a link to a WAP address. It can be delivered over WAP or SMS (Simple Message Service) bearer. On receiving WAP push messages, WAP enabled handsets can automatically give the user the option to access the WAP content [11].

The push functionality is especially relevant to real-time applications that send notifications to users. Without the push functionality applications would require the devices to poll application servers for new information or status. In wireless environments such polling activities would constitute inefficient and wasteful use of resources of wireless networks. WAP’s push functionality provides control over the

lifetime of pushed messages, store and forward capabilities at the push proxy and control over bearer choice for delivery [7].

The WAP push service directs the end user to a WAP address where particular content may be stored, which is ready for viewing or downloading to the handset that enhances usability. Operators and content providers can utilize push data transfer to deliver content that is relevant to user groups, instead of relying on the traditional pull model. WAP push advances existing messaging models (SMS, Smart Messaging) by integrating them into the WAP application environment. In consumer segment, users can subscribe to content they are truly interested in or to receive push messages that add value (promotions and discounts). The push message can inform them to follow links to more details or to complete a transaction. In a corporate segment, the company can send important news and users can access vital, time-critical information easily.

A push operation in WAP is accomplished by allowing a PI to transmit push content and delivery instructions to a PPG (Push Proxy Gateway), which then delivers the push content to the WAP client. The PI is typically an application that runs on an ordinary web server. It communicates with the PPG using the Push Access Protocol (PAP). The PPG uses the push Over-the-Air (OTA) protocol to deliver the push content to the client. The architecture of push can be thought of similarly to that of SMS, except that with push, the PPG is found in the middle rather than the SMSC. Push can be sent over SMS or GPRS [16, 19].

There are a number of push service types; SI (Service Indication) and SL (Service Loading) are some of this. SI allows you to have a web page links sent to you and placed in your mobile phone inbox. But in SL, the content of a page can be sent to the user that can be interfering, non-interfering or to be stored in the phone cache.

WAP push service enables users to receive timely information based on preferences, and giving users a feeling of control and makes service discovery easier. The WAP push service can include advanced push message management (high, low, cache). This makes easy navigation with embedded links, encourages browsing, and makes possible to push a better content than text with SMS [10].

It is possible to develop different kinds of application that give WAP push services [16]. Some examples of WAP push services are: -

- Time Triggered Push: to deliver information on timely basis
- Content Triggered Push: for example, the deliver of information based on the current market price changes or availability of item list with their current price
- Frequent Flyer Programs: to deliver information that need to be announced frequently for the customer, for example airline flight schedule notification and currency conversions
- Community Services: to deliver notification to a community members, like dating notification when match is found, the scores of the game and notice to family members
- Auction Services: to deliver bid notices like outbid notification and successful bid

2.4. WAP Push Application Deployment Examples

These days there are many operators who provide a WAP push application service around the world. This is because of the mobile telephone subscribers shows an interest on the use of mobile Internet. Here we want to see some operators, such as Sprint PCS (from America), LGT (from Korea) and Telesp (from South America, Brazil), that have been giving a mobile Internet service using a WAP push standard [11].

Sprint PCS has proven that push-enabled applications can make a significant difference in adoption and usage. Sprint continuously improves the services it offers to subscribers and deploys technologies that will improve the user experience. One example is AOL Instant Messenger. Within two weeks of implementation, the push-enabled service generated over 1 million minutes of use. Push allows the subscriber to be alerted when friends are online and when the subscriber receives a new instant message. More importantly, it allows the subscriber to immediately respond to the message with one or

two clicks. Push enables an experience on the mobile handset that is very close to that of the desktop.

LGT is a Korean operator who has deployed the Openwave Mobile Access Gateway and leverages its push capabilities to launch a wide variety of services. LGT has researched the push market extensively and push will continue to play a significant role in its success, accounting for a significant volume of data transactions. Expected benefits include “pull” traffic and increased usage of related services. It has over 4.3 million voice subscribers; of whom over 70% are WAP users. LGT has achieved a response rate much greater than 10% to push notifications. Push services offered by LGT include Email notifications, news, details and entertainment applications. This proves that subscribers respond to a well-designed push program that delivers truly valuable content and applications.

Telesp is a relatively new South American mobile operator that has achieved one of the world’s fastest adoption rates of mobile Internet services within the first 12 months of deployment. Telesp has achieved 22% mobile Internet penetration by delivering personalized services targeted at a specific segment, attracting prepaid users and delivering compelling applications in a short space of time.

In conclusion, Sprint, LGT, and Telesp have demonstrated push to be a powerful application capability that increases mobile Internet adoption and revenue. Push provides significant improvements in the usability of applications, allowing proven Internet applications like instant messaging to be effectively replicated on the mobile Internet. It takes mobile person-to-person messaging to a new level of usefulness, and it enables developers and operators to deliver valuable content, information, and applications to subscribers anywhere and anytime.

2.5. Markup Languages and Ethiopic Content Display

The introduction of an Ethiopic character set and character code conversion algorithms [8] provide options to develop web sites, which display a web content in Ethiopic script.

Ethiopic characters are introduced since Unicode standard version 3.0 [2]. There is an introduction of using transliteration, conversion and a Unicode encoding of Ethiopic character sets to marked-up web pages in Ethiopic. But the majority of Ethiopian web pages is not utilizing it to display Ethiopian character set. This shows that the usage of Ethiopic encoding for appropriate applications is at its infancy stage.

The markup languages such as HTML and XML has a support for a Unicode character set, and Ethiopic character sets are included in Unicode standard [2]. This indicates that, there is the possibility of developing Ethiopic content web pages using HTML and XML. For document and data interchange over the Internet and the World Wide Web, it is possible to develop web pages that contain an Ethiopic character set whether developed using HTML or XML.

Revision have been done on the XML standard to support Ethiopic based XML element types and attribute names in Ethiopian local languages, which uses Ethiopic Script [8]. This also shows the possibility of developing an Ethiopic content WAP page using Ethiopic scripts. But there is no work done so far in developing an Ethiopic content WAP page using Ethiopic scripts.

However, the above effort indicates that developing a WAP page using WML (XHTML MP) for Ethiopic content is possible. Because WML and XHTML MP are a markup languages based on XML specifically designed for mobile devices.

To our knowledge, there is no work done so far to develop a WAP page or a WAP application with an Ethiopic content. So, the task of this project is to develop an Ethiopic content WAP page using an appropriate markup language that is browsable by mobile devices, and to show the practicability of the development of an Ethiopic content WAP pages for mobile devices.

To demonstrate this work with practical application of Ethiopian interest, we have chosen a delivery of marketing information to some of cooperative societies under the FCC. Thus, analyzing the requirements of marketing information in the selected application area was a requirement that the documentation proposed for [23].

The on going effort for establishing a nation wide network system has been considered as a major input of information. We have observed that the country intended system to create a computer network that requires a huge infrastructure development and a high financial and human resources requirement.

3. Requirement Identification

FCC is a government institution, responsible for registering and supporting cooperative societies organized at federal level, and which conducting research, rendering training and other technical support for the cooperatives. Under FCC, there are branch offices such as, the RCB (regional Cooperative Bureau), ZCO (Zonal Cooperative Office), and WCO (Woreda Cooperative Office) who provide services for the Cooperatives.

One of a service that was intended by the FCC is to provide marketing information for their cooperatives. But so far it is rarely provided for the cooperatives. The FCC's information system including the marketing information system was analyzed and designed by the consultant for implementation [24].

Biz Soft Plc (the consultant) have taken the task and performed the analysis, design and developed a web-based marketing information system. However, the system is not implemented so far to provide services to cooperatives [24].

The current situation shows that market economies run on the basis of information. Getting better marketing information (i.e. the accuracy and timeliness information) helps to make a better decision, this leads to greater economic growth. So the development of efficient and equitable cooperative marketing information is a critical component. It improves the performance and product availability, thus the producer can get a better price for their product at least in their local market.

The timely and accurate market information improves awareness of the producers on the prices of products in various market places throughout the country. Among the many advantages of this market information are [23]:

- Encouraging product flows from relatively surplus to relatively deficit areas, helping to stabilize prices for consumers
- Improving farmers decisions and confidence regarding what to plant, how much to invest, and where & when to market their product

- Promoting a more competitive marketing system, which will benefit both producers and consumers by improving quality of product

In the current situation, the marketing information service and other services are provided by the WCO for the cooperatives. But cooperatives and unions are not relying on WCOs, because WCOs are not proactive, in collecting and disseminating marketing information. Thus marketing is based on inadequate market information. The already adopted means of getting marketing information is using radio and TV programs, personal report or visiting the market of selected places. As a result:

- The information is generally limited to a very small number of markets
- Visits far away markets involves significant costs
- Poor and remote farmers and cooperatives may be unable to access information on time

Based on the information in [24], the agricultural products are categorized in to groups. Among the many categories, what is considered in this work are:

Category	የምርት መደብ
1. Cereals	የብዕር ሰብል
2. Stalk	የአገዳ እህል
3. Coffee	ቡና

The types, standards and origins of the product should be specified while delivering the marketing information given in Table 3.1, Table 3.2 and Table 3.3. This information category will be used to specify the current market information in the different market places.

Table 3.1 Cereal Products (የብዕር ስብል ምርቶች)

Product name (የምርት ስም)	Type (ዓይነት)	Standard (ደረጃ)	Origin (የምርት ቦታ)
Teff (ጤፍ)	White (ነጭ)	Magna (ማኛ)	ምንጃር አድአ በቾ ጎጃም
	Mixed (ቅልቅል)	Sergegna(ሠርገኛ)	
	Red (ቀይ)	Key (ቀይ)	
Wheat (ስንዴ)	Durum (የመኮሮኒና ፓስታ)		
	Bread (የዳቦ)		
Barely (ገብስ)	Yebira bikl (የቢራ ብቅል)		
	Yetela (የጠላ)		
	Yebeso (የበሶ)		
Dagusa (ዳጉሳ)	White (ነጭ)		
	Black (ጥቁር)		
Oats (አጃ)			

Table 3.2 Stalk Products (የአገዳ አህል ምርቶች)

Product name (የምርት ስም)	Type (ዓይነት)
Maize (በቆሎ)	Maize (በቆሎ)
	Popcorn (ፈንዲሻ)
Sorghum (ማሽላ/ዘንጋዳ)	White (ነጭ)
	Red (ቀይ)
	Black (ጥቁር)

Table 3.3 Coffee (ቡና)

Product name (የምርት ስም)	Type (ዓይነት)	Standard (ደረጃ)	Origin (የምርት ቦታ)
Coffee (ቡና)	Washed (የተጣራ)	Export (ለውጪ ገበያ የሚሆን)	Sidamo (የሲዳሞ) Kafa (የከፋ) Welega (የወለጋ) Limu (የሊሙ) Yirgachefe (የይርጋጨፌ)
	Unwashed (ያልተጣራ)	Reject (ለውጪ ገበያ የማይሆን)	

From these categories of products, we observe that any application of marketing information delivery should be designed in the same way. Further more the marketing information should include:

- Market place
- Type and standard of the product
- The measurement unit, and
- The unit price

4. Tools and Technologies Used

4.1. Overview

WAP applications are often developed on personal computers. While developing the application on personal computers, we need to test it as if we are using the real WAP devices, so we need a simulator. A simulator is a term used to describe a WAP device implemented in software, and as such is not a physical device, like a mobile phone for example. Simulators arrive as part of many SDKs and allow for local WAP development, saving time spent on air with a real WAP device in order to test WAP applications. Simulators can also be used with online content, connecting directly to a given website using HTTP.

There are several types of SDK (Software Development Kit) for developing a WAP push and pull applications. Openwave and NMIT (Nokia Mobile Internet Toolkit) SDKs are widely used WAP push application development environments. These systems are freely downloadable from the websites of Openwave[1] and Nokia[2].

Using Openwave SDK the push applications are easily done with the PushIT tool included in the Openwave WAP Push library package. PushIT has a general user interface for building PAP messages and sending it to handsets. In order to simulate an end-to-end push environment using the PushIT tool, we need to access a real Push Proxy Gateway. The real Push Proxy Gateway can be configured to access PushIT tool, and the client ID by which the SDK will be identified while targeted for push messages. We can get access to the free real Push Proxy Gateway that is provided by Openwave, by registering online and getting an account ID. However, for developing and testing a push application using an Openwave SDK, an Internet connection is required to get access to the Push Proxy Gateway, for push messages [32].

[1] http://www.Openwave.com
[2] http://www.Forum.Nokia.com

The NMIT is a development environment, using which different Internet applications can be developed for mobile device. Some of the contents that are created by NMIT are browser content, MMS messages (Multimedia Messaging Services) and push messages. Furthermore NMIT enables to preview content on supported phone SDK [27].

In addition to NMIT, Nokia provides NMBS (Nokia Mobile Browser simulator) and NWGS (Nokia WAP Gateway Simulator). The NMBS is a simulator, which can display any WAP content that is developed using NMIT by accessing locally or through a WAP connection [26]. WAP connections may be made through either a WAP gateway server or through NWGS. The NWGS is a single user WAP gateway, which can be integrated with NMIT to access WAP content from the web server [28].

From the development environments available freely, I have chosen the NMIT with NMBS and NWGS, for the development of the WAP push application. The main reason for selecting NMIT is that NMIT enables to test a WAP push application with out the requirement for Internet connection to integrate the WAP gateway for testing the application. But Openwave SDK requires Internet connection for integration with the gateway, for testing the WAP application [28, 32].

To place a WAP pages and test the accessibility of WAP application, we choose the Apache tomcat server, which is freely downloadable open source software.

4.2. System Requirement

The NMIT and the NMB (Nokia Mobile Browser) SDK requires the following minimum hardware and software requirements. The software requirements they need are Windows 2000 Service Pack 3 (Windows XP Professional Service Pack 1) or later, and public JAVA Runtime Environment JRE 1.4.1_02 or later. The hardware requirements are at least 50 MB disk space, 256 MB RAM, 64K color 16-bit resolution, 500 MHz or faster Pentium processor, and keyboard and mouse [26, 27, and 28].

4.3. Nokia Mobile Internet Toolkit

NMIT consists of a set of editors that you can use to create various types of mobile Internet content (Figure 4.1). NMIT lets us to push WAP content to multiple phone SDKs. Phone SDKs are emulated mobile phones that can be installed separately. NMIT detects installed, supported phone SDKs at startup and lists these in its SDK Control Panel. You can send any content created on any supported phone SDK by simply clicking a Show button within the editor. Many NMIT editors are used for creating XML-based content types defined by DTDs (Document Type Definitions). These editors employ content validation to check content against a DTD, and they provide features for easily selecting elements and attributes for insertion based on current cursor position. In addition, NMIT provides a DTD Manager through which you can import new DTDs for use by NMIT editors [27].

The types of contents created by NMIT are categorized as:

- Browser content
- MMS (Multimedia Messaging Services) messages
- Push messages, and
- DRM (Digital Rights Management) messages

From all this content, the browser content and the push messages are the main contents for WAP application developers.

The browser editors are used to create:

- WML1.3 Deck (a Wireless Markup Language document that supports the WML 1.3 DTD),
- WMLScript (a content which enables to add programmatic logic to a WML deck)
- WBMP (a Wireless Bitmap image and enables conversion of existing images from GIF and JPEG formats to WBMP format)
- XHTML MP (an XHTML document based on the XHTML-MP DTD)
- CSS (a Cascading Style Sheet (CSS) that contains formatting style definitions that will be applied to the elements specified in an XHTML document), and
- XHTML-MP + CHTML (an XHTML document based on the XHTML-MP DTD with additional element and attributes taken from Compact HTML).

The push content editors are used to create:

- SI content (a push message which is sent to notify a user of the availability of new content)
- SL content (a push message which is sent to force a user agent running on the client device to download new content, without notifying the user)
- CO (Cache Operation) content (a Push message that is sent to invalidate specific content in the user agent cache), and
- Multipart message (a kind of Push message consisting of more than one part, each of which is separately processed by the user agent).

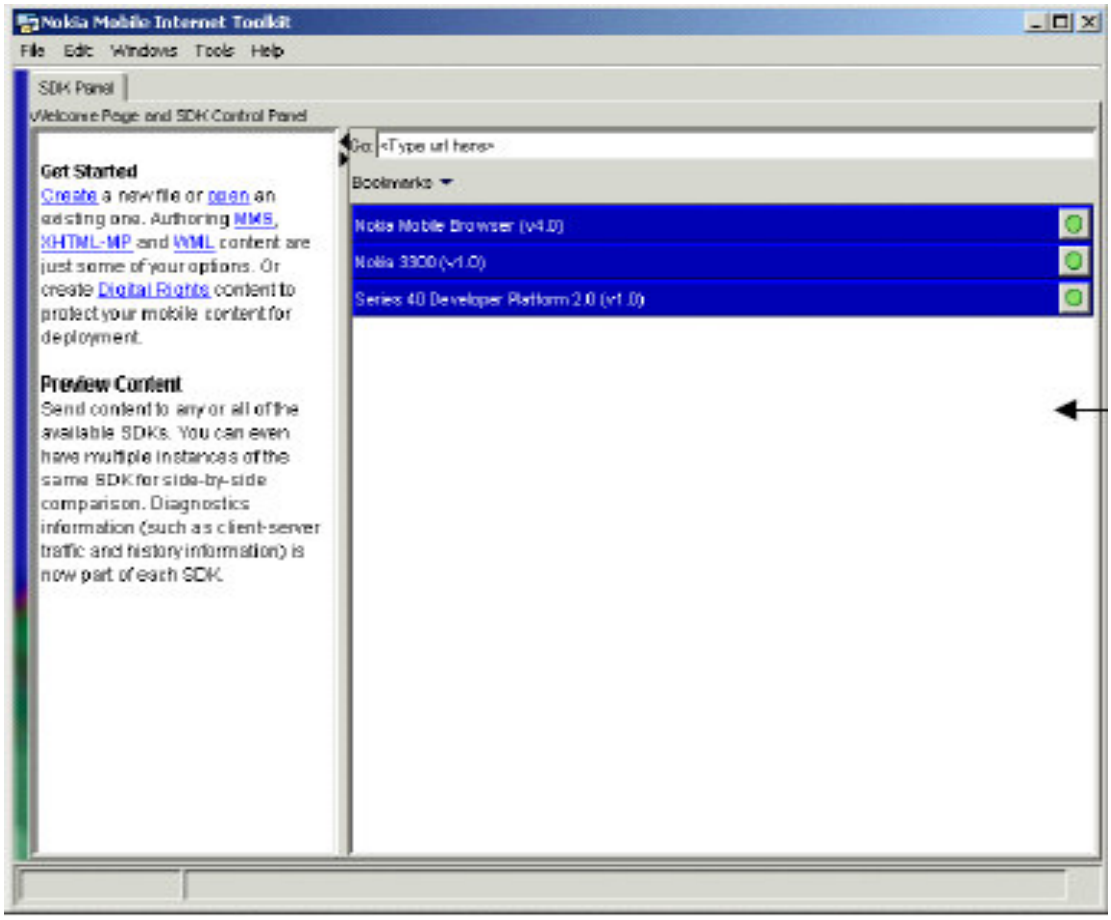

Figure 4.1 Nokia Mobile Internet Toolkit

4.4. Nokia Mobile Browser 4.0 SDK

Nokia mobile browser (NMB) is a mobile phone SDK that includes a mobile Internet browser for browsing both mobile Internet content (through a WAP connection) and local file content (Figure 4.2). It fully supports the content authoring features of NMIT 4.0 and can be used to display both XHTML and WML content, as well as push messages. It is designed for use by mobile Internet content developers who wish to display Internet content on a mobile handset emulator. This content may be local file content; content accessed over the Internet, or content developed within NMIT and pushed directly from within the NMIT environment [26].

NMB is also a tool intended for mobile Internet content developers who wish to preview how their content will look before it is ultimately deployed on a mobile phone handset. Using NMB, content developers can display any mobile Internet content developed using NMIT, as well as local file content and content resident on Internet servers and accessed through a WAP connection. WAP connections may be made through either a WAP gateway server or through Nokia's WAP Gateway Simulator (NWGS).

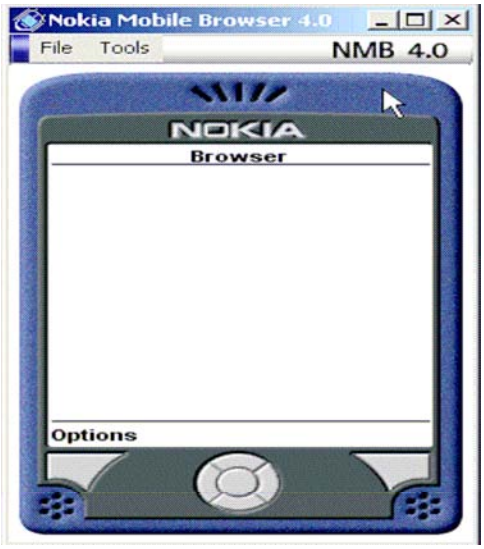

Figure 4.2 Nokia Mobile Browser Simulator

NMB uses the Nokia Mobile Browser software, which has been developed by Nokia for deployment on actual phone handsets. But it is not designed to reflect the functionality of

any particular handset but rather an extensive range of current and evolving technologies of interest to mobile Internet developers.

NMB supports different character set including UTF-8. So it can use different font families based on the XHTML content that is specified in the CSS's font family property. If the content specifies font family property, the NMB uses the corresponding true type font that installed in the mobile phone. Otherwise the NMB uses the Arial font family, which is the default [26].

4.5. Nokia WAP Gateway Simulator

The NWGS is a separate application, which can be integrated with NMIT. It is a single-user WAP Gateway based on the multi-user Nokia Active Server. When installed on your computer, NWGS enables you to access the mobile Internet through the phone SDKs that you use in conjunction with NMIT. NWGS provides a subset of the features provided by Nokia Active Server [28].

NWGS is a WAP 2.0 compliance with support for WML, WMLScript, XHTML, CSS, and Push message content types. It includes a decoder for decoding incoming requests from WAP client user agents, such as mobile phone emulators (SDKs), so that these can be forwarded over the HTTP protocol to Internet servers. It also includes an encoder that is used to encode server (HTTP) responses before sending these back to requesting clients.

NWGS uses UDP/IP bearer adapter. This adapter enables communication between NWGS and other client user agents running in a local area network, such as NMIT and phone emulators. NWGS does not support other bearer adapters supported by Nokia Active Server as these are designed to enable radio communication between devices and the mobile Internet.

It is possible to configure NWGS using the GUI of its application window (Figure 4.3). We can stop and start traffic, configure proxy and cache settings, and view and configure log file settings. For example depending on your network configuration, you may need to

specify an HTTP proxy server. This would be the case, when your computer was located within a corporate Intranet that used an HTTP proxy server as a gateway to the Internet.

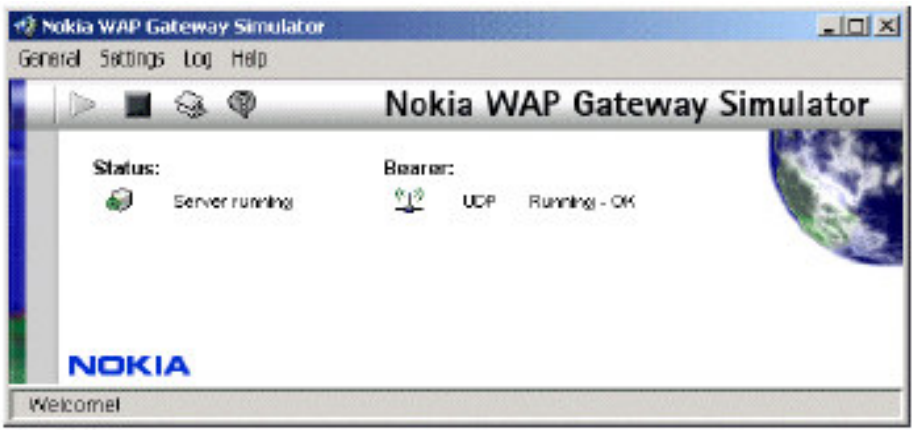

Figure 4.3 Nokia WAP Gateway Simulator Administration view

4.6. The Sadiss Ethiopic text Editor

Sadiss is a Unicode text editor for Ethiopic (Figure 4.4). It is developed using java and needs JRE (Java Run time Environment) 1.4 or later to run. It can be localized for other scripts with modest changes on the source code [30].

Generally Sadiss supports only plain text, for the text that uses Unicode, it provides two options, UTF-8 and plain Unicode. Internally, Sadiss works with Unicode character, but it converts the text from Unicode to UTF-8 when it writes to a file and also performs conversion from UTF-8 to Unicode while reading from a file.

Currently, Sadiss supports the KWK[3] keyboard layout for Ethiopic. This keyboard layout is a modified version of the WashRa KWK keyboard system. It maps each Ethiopic character to a standard keyboard. We can refer the table that shows how can we map an

[3] An acronym stands for Kidane Wolde Kifle, who was a linguist, theologist and writer

Ethiopic character with the keyboard layout that is found in the help part of the application.

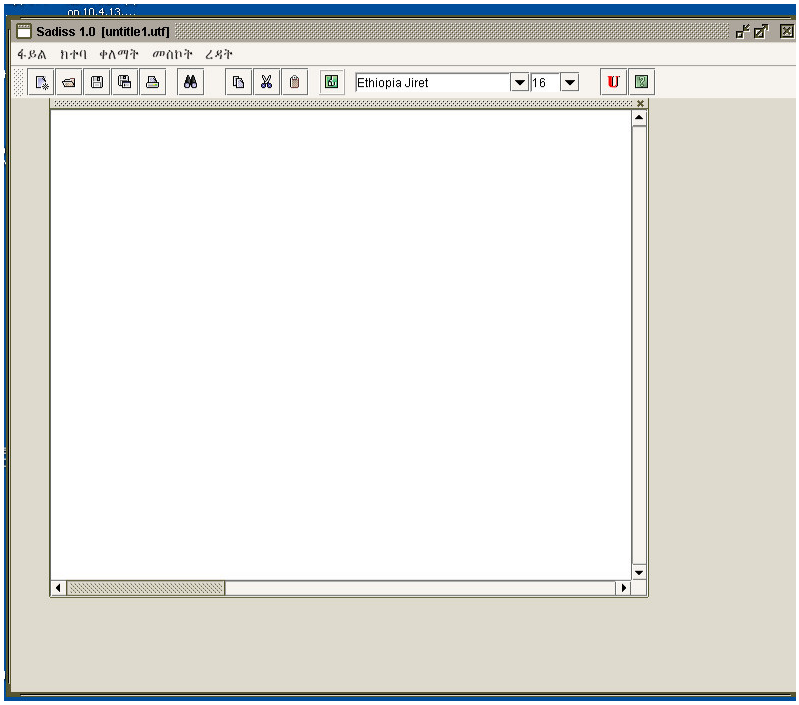

Figure 4.4 Sadiss (Unicode Ethiopic text editor)

4.7. Apache Tomcat Server

Tomcat is a freely available reference implementation of the Java Servlet and Java Server Pages (JSP) specifications. It is developed and released under the Jakarta project as the official JSP 1.1/Servlets 2.2 reference implementation by the Apache software Foundation [31].

The mission of the Jakarta Project was to provide commercial-quality server solutions based on the Java Platform that is developed in an open and cooperative fashion. The product of this project, Tomcat, is a reference implementation of the web page specifications, which can run standalone as well as integrated into the Apache Web Server. This reference implementation provides an operational definition for the Enterprise JSP and Servlet application-programming interfaces. Tomcat available to any company or developers to be used as web servers, development tools, and to create static, dynamic, interactive Web sites.

By using all the above-indicated tools, it is possible to develop a WAP application in Ethiopic content. The application can be placed in the tomcat web server, which enables to show the possibility of accessing using mobile phones through the WAP gateway.

5. Architecture and Configuration of the System

5.1. The WAP Push Application Architecture

The WAP protocols are designed to operate over a variety of different bearer services, including short message, circuit-switched data, and packet data. Actually the bearers offer differing levels of quality of service with respect to throughput, error rate, and delays. But the WAP protocols are designed to support these varying levels of service [19].

Since the WDP layer provides the convergence between the bearer service and the rest of the WAP protocol stack [1]. Its specification lists the bearers that are supported and the techniques used to allow WAP protocols to run over each bearer. The list of supported bearers will change over time with new bearers being added as the wireless market evolves.

In addition to the set of protocols contained, WAP has the following architectural components, such as the micro browser in the mobile devices, the WAP gateway (proxy) that specially resides on the carriers network (but can also placed with the web server), and the web server that hosts the WAP content [18]. The WAP push application general architecture is as shown in Figure 5.1.

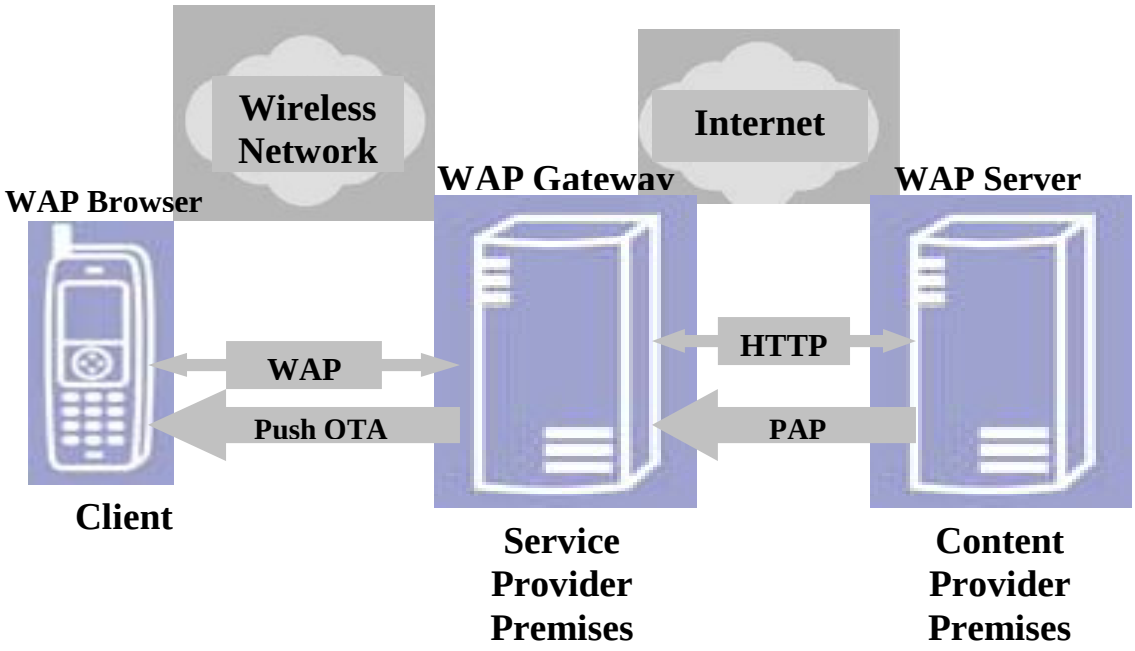

Figure 5.1 General Architecture of the WAP Push Environment

In the WAP push application development; there is an application server in which WAP content and push library will be stored [18]. The web server will be located at the premises of the content provider, because it is the place from where service is rendered. The server should hold both the WAP application content and the WAP push library. A PI, an application that runs on server, performs a push operation by submitting push content (it can be a URL of a WAP application) with other delivery instructions to a WAP gateway. It is communicating with the WAP gateway using the PAP over HTTP to push the content [11].

Since the OTA contains a number of bearer adaptations as specified by WAP, thus by using the available bearer services we can provide the WAP push services for mobile terminals.

The WAP gateway can for example be placed in the Ethiopian Telecommunication Corporation (ETC) premises. Because using the same WAP gateway, ETC can give service for different WAP application. Currently the SMS is provided by the ETC [25] and SMS is one of a bearer services that are adapted by the Push OTA. So it is possible to use the already available bearer service as a delivery channel.

The WAP gateway performs the needed protocol and format conversion to transmit content to a mobile device. Once configured, the gateway dynamically delivers the push content to wireless terminals (devices) through the wireless network. The gateway uses the Push OTA protocol to deliver the Push content to the wireless terminal [18].

The client or the mobile phone should be WAP enabled. As a result, the WAP browser on the mobile phone becomes capable of displaying the WAP content sent from the WAP server.

5.2. Configuring the System

The configuration of the development environment starts by configuring the MIME (Multi purpose Internet Mail Extension) type of the application in the web server and the WAP gateway. The MIME type is the application type that the web server defines to be recognized by the web browsers [28].

In this project, Apache server is used for the placement of the WAP application. Hence the web server should tell the WAP browser the kind of document it should expect, by defining its MIME type. When the web server needs to forward some content to the browser, it should specify the type and content of application. So before using a new application type, we need to configure its MIME type in the server.

The WAP gateway we used for the demonstration of the WAP push application is NWGS. It can work with NMBS and NMIT and also with the server, for the wireless Internet browsing.

The NMBS can use a NWGS or other WAP gateway to fetch Internet content. The NWGS can directly connect NMBS to the Internet, if there is direct Internet connection to your computer. But, if your computer has access to the Internet through a proxy server, you will need to configure NWGS to use this proxy server. By default the NMBS is configured to use the NWGS whose IP address is 127.0.0.1. Therefore it is possible to browse an Internet content using NMBS through NWGS. The NMBS can also connect to the Internet through any other WAP gateway, but it should be configured.

6. WAPE (Wireless Application Protocol for Ethiopic), the Prototype

6.1. Markup Languages used for Implementation

The main types of content that mobile browsers support are WML and XHTML MP. WML is a markup language that is developed for writing WAP applications that can be browsed by mobile browsers. XHTML MP is a standard used for all browser-based services using mobile phones introduced with WAP 2.0. So WML and XHTML MP can be used for WAP application content development.

The XHTML MP, in addition to content development, it supports CSS that describes how documents presented on screen by the browser. The support of CSS enables application developers to create browser specific versions of the same content easily, simply by creating style sheet. Therefore XHTML MP with CSS can be a choice for Mobile web application development [9, 22].

The prototype, which is used to show the possibility of developing an Ethiopic content WAP application, is developed using XHTML MP and WML. We will XHTML MP to develop WAP pages that needs more features that cannot be implemented with WML. We used WML to place different presentation in a single document. Because WML makes possible different content to be placed on different cards, and create a link between them in a single WML file (deck). Since the WML file is fetched only once, and stored on the cache of the phone, so that content of the card can be retrieved from the cache.

6.2. The Prototype

A push message is unsolicited message sent by a server to a client. A server application called a PI initiates the sending of a push message. It sends this message via the PAP to a PPG. The PPG extracts the relevant client-bound content, encodes and then prepare the required content type header and any other included headers (all un encoded). There after using the WAP protocol it sends the content over a wireless network to the destination mobile handset.

The mobile handset decodes the message, recognizing the content type as push-related and dispatches to the content handlers. For example, SI content is dispatched to the handset inbox. The user of the handset then views the message in the inbox and can retrieve the content by selecting the link.

In the NMIT application development environment, there is no PI. Therefore, there is no use of PAP or PPG, and hence no messages traverse in a wireless network. Instead there is a Nokia-internal protocol whereby messages you create using NMIT Push editors can be delivered to phone SDKs that run on the same local machine. The aim of this is, to enable us to work within NMIT to construct a push message just as a PI does. It encodes and includes the appropriate headers, to deliver the push message to phone SDKS in the same manner as a PI, which delivers Push messages in the wireless network environment. The delivered messages therefore are structurally similar to those received over the network [26].

NMIT provides Push message editors, which can be used to create push messages and deliver push content on a phone SDK. This is similar to how Push messages are deployed over the network to mobile handsets that can show the reality. Because, the push message can go through the WAP gateway and which is browsable from the phone SDK.

Phone SDKs expect to receive only WAP-encoded Push message content. Therefore, messages that you push to phone SDKs are always encoded by NMIT.

From the push application editor provided by the NMIT, we used SI and SL editors. Both SI and SL enable us to create push messages and uses different approaches to deliver the message.

The SI Creates a message intended to be delivered to the phone SDK's inbox, not to the phone browser. The interface of the SI editor contains the following text boxes:

- Content type: to specify the type of push message content
- Text message: to specify the label of the push content that is used as a notification to the user, to show new content is available for downloading.
- Href: to specify the URL address of the content
- Active: to specify the level of urgency (priority) of the message to be adopted by the user of the phone SDK.
- Char. Encoding: to specify the encoding used by the push application

The editor has the Push and Store binary (used to store the message as encoded format) buttons. The Push button is used to push the specified message to the running phone SDK.

The push application developed using the SI editor is shown in Figure 6.1. When the application service provider intends to push the message to the phone SDK, he presses the push button. Thus the text message with the URL of the content placed in the inbox of the Phone SDK. The user of the service can access the message from the inbox by selecting the supplied link (Figure 6.5).

When a SI push message is sent to the user, the message is not shown to the user. But the user can check for the message from the inbox. To go to the inbox, the user can select the options command displayed on the bottom left corner (Figure 6.3). When the user selects the inbox from the option (Figure 6.4), he/she would get list of SI messages (Figure 6.5). So the user can select one to fetch the content from the server.

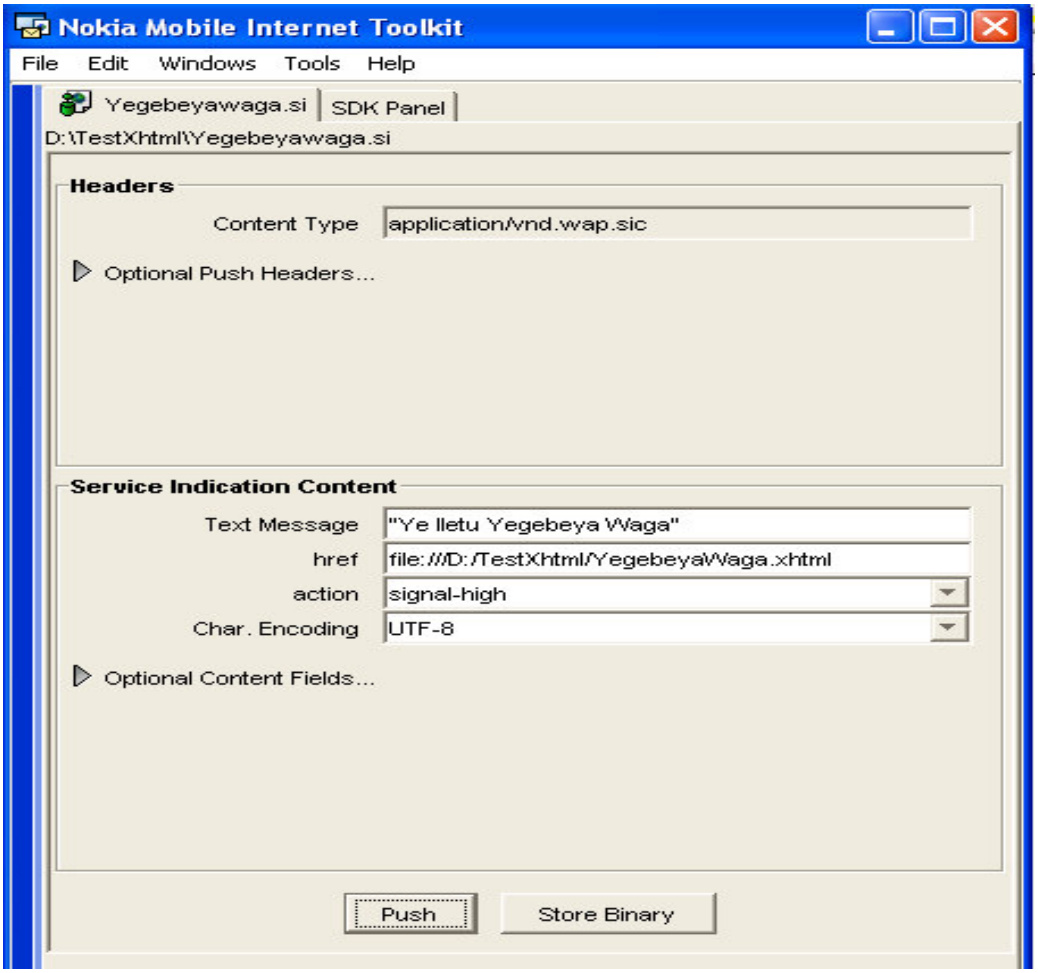

Figure 6.1 A push application developed using SI editor

The SL is another editor used to create push message. This type of push message is capable to force the client to view new content by interrupting what he/she is doing.

The interface of the SL editor contains the following text boxes:

- Content type: to specify the type of push message content
- Href: to specify the URL address of the content

– Active: to specify the level of urgency of the content to be adopted by the user of the phone SDK. If the value specified is cache, the message should be place in the phone cache.

Figure 6.2 shows the push application that is developed using the SL editor. When the push message is pushed to the phone SDK, the message content is displayed as shown in Figure 6.6 by interrupting the current state of the user.

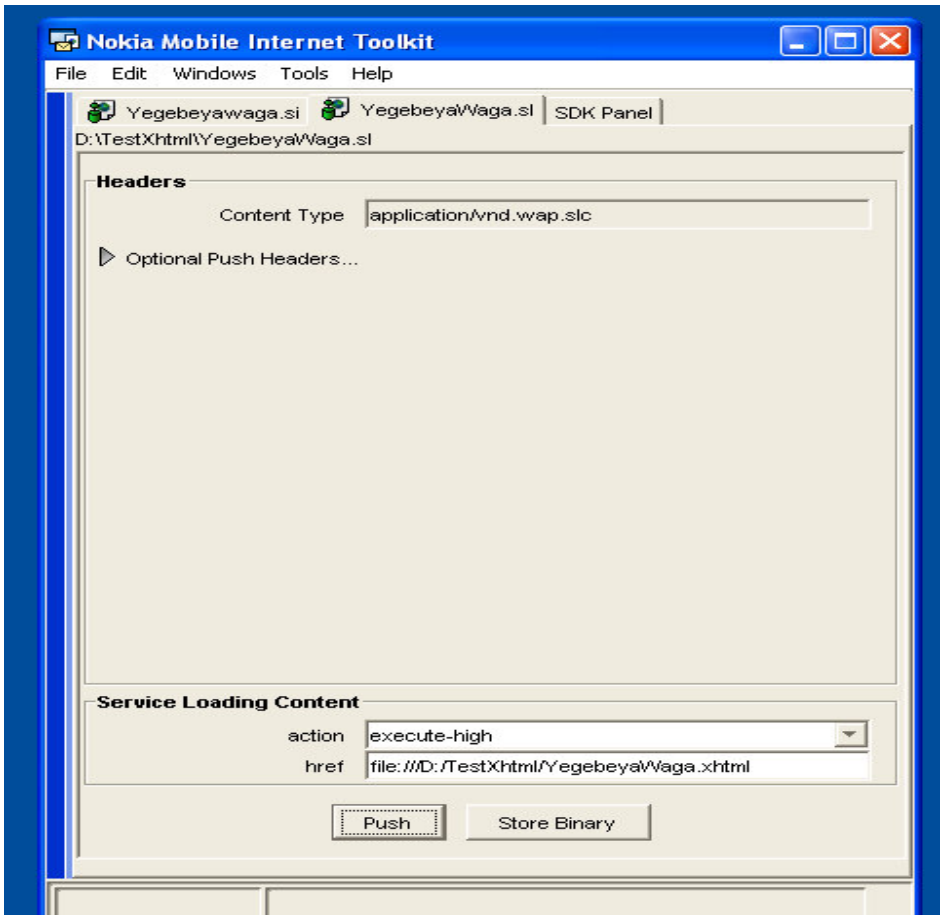

Figure 6.2 A push application developed using SL editor

Figure 6.3 A phone SDK before accessing the push message sent using SI

Figure 6.4 The phone SDK options

Figure 6.5 The push message sent using SI placed on the phone inbox

When a SL push message sent to the user, the content of the message is shown to the user as shown in Figure 6.6. The user can select the link to browse the content or select back to disregard the message.

Figure 6.6 The push message content displayed on the phone browser sent using SL

If the user needs to access the current market information of the specified product, she/he can retrieve by selecting the successive link of the required product.

To demonstrate the prototype implementation using the product categories specified in chapter 3, we consider the case of the Sidamo coffee product. First the user needs to select one from the product category (“**የብዕር ሰብል**”, “**የአገዳ አህል**”, “**ቡና**”) from the specified list (Figure 6.7). If the user selects “**ቡና**” the user gets another list (product origin list) (Figure 6.8). To know the current market information about Sidamo coffee product, the user needs to select the “**የሲዳሞ**” link from the specified list. So the current market information of the specified product for different market place, product quality, unit of measurement and unit price in tabular format as shown in Figure 6.9.

Figure 6.7 The list of product categories displayed on phone SDK

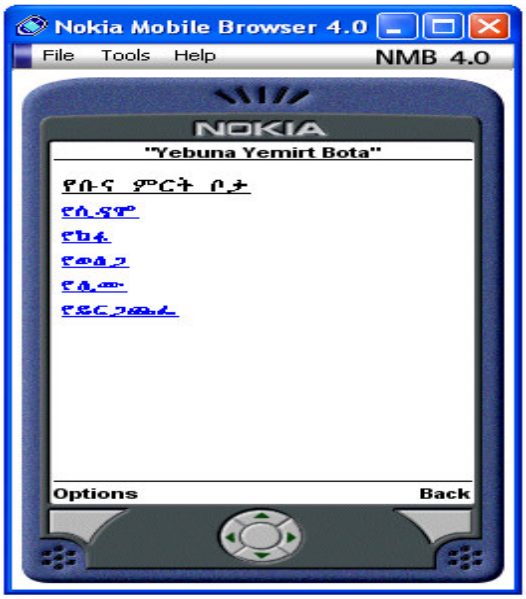

Figure 6.8 The list of product places as displayed on phone SDK

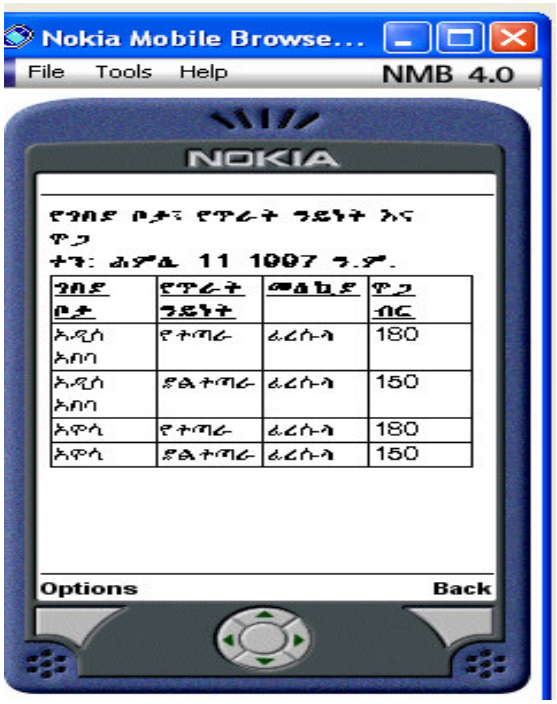

Figure 6.9 The current market information as displayed on phone SDK

7. Conclusion and Future Work

7.1. Conclusion

Nowadays, the retrieval of information from Internet from anywhere and at any time using the mobile device is an interest of every body. Accessing web content anytime from anywhere is becoming practical using mobile devices based on WAP standard. Delivery of information using local language is also another critical issue for many users.

In this work, the necessary standards, tools and configuration that are required for developing WAP application are investigated. In this regard protocols and different tools, such as WAP, PAP, OTA, WML, XHTML and soon are identified and detail studies are conducted.

The possibility of using local languages based WAP application development is also explored. In this regard the fact is that, Ethiopic is Unicode supported. Therefore the possibility of developing WAP applications for Ethiopic content is facilitated. Thus the necessary encoding scheme based on Unicode that supports an Ethiopic character sets are identified.

Since testing a WAP application under a real environment is not possible, an emulator environment is identified and used. In which case NMIT, NMB and NWGS are used to test the possibility of WAP application for Ethiopic content.

To demonstrate the technology under a real application environment, the requirement of market information delivery under the FCC is identified. Based on the result of this requirement identification, an application for agricultural marketing information delivery is developed.

We also studied that for market information delivery using mobile communication, Push service is relevant because they use the limited bandwidth efficiently, i.e. communication occurs if there is available information. But in the case of pull-based communication, the user performs periodic checks for information - even if no new content is available.

From this prototype development, we can see that there is a possibility of developing a mobile Internet application using Ethiopic content. Depending on the customer need and their mobile phone capability of browsing WAP application, an Ethiopic content WAP application can be provided by the concerned service provider.

7.2. Future Work

As a future work, this project may extend to:

- Real Ethiopic content WAP application development for different application.
- Developing a dynamic and interactive Ethiopic content WAP site, which is accessible using mobile phone.

References:

1. WAP Forum, “WAP 2.0 Technical White Paper”, January 2002, www.wapforum.org, visited March 10, 2005

2. Alan Wood’s Unicode Resources, “Test for Unicode Support in WEB browsers: Ethiopic”, April 9 2003, http://www.alanwood.net/unicode/ethiopic.html, Visited March 12, 2005

3. Jack’s Hack, “Using International character sets with the openwave Mobile Browser”, April 2002.

4. Forum Nokia, “How to Display Chinese Characters in WAP and MMS”, version 2.0, September 24, 2003

5. Peter Rysavy, “WAP: Untangling the Wireless Standard, November 2000”, http://www.nwc.com/1123/1123ws1.html, visited March 15, 2005

6. S. Jadeja, “White Paper on Wireless Application Protocol”, http://www.dwge.com/case_studies/white_papers.htm, visited March 20, 2005

7. Openwave Systems Inc., “WAP Push Technology Overview”, May 2002, www.openwave.com, visited March 20, 2005.

8. Samuel Kinde Kassegne, Bibi Ephraim, “Need for Revision of XML 1.0 to account for Localization Issues with Particular Emphasis on Ethiopic Script and Writing System, January 2002”, http://www.digitaladdis.com/sk/ETXMLFinal.pdf , visited March 10, 2005

9. Nokia Corporation, Nokia mobile phones “ White Paper -Next Generation Mobile Browsing Rich and colorful browsing for mobile users”, 2003, www.Nokia.com , visited march 18, 2005.

10. Openwave Systems Inc., “Comparison of WAP Push and SMS, April 2002, http://developer.openwave.com/docs/wappush_vs_sms.pdf, visited May 2005

11. Openwave Systems Inc, "The Value of WAP Push", December 2001 http://www.openwave.com, visited March 20, 2005.

12. Fouzi MOUSSA - Sahad CHOUF, "Wireless Application Protocol", January 26 2001, http://dessr2m.adm-eu.uvsq.fr/thewappresentation.pdf, visited May 2005

13. Uwe Hansman, Lothar Merk, Martin S. Nicklous, Thomas Stober, "Pervasive Computing: the Mobile World", Second edition

14. Jochen Burkhard, Dr. Horst Henn, Stefan Hepper, Klaus Rintdorff, Thomas Schack, "Pervasive Computing Technology and Archtecture", Greate Britain 2002.

15. Forum Nokia, "Getting Started With WAP Push", Version 1.1; August 11, 2004, http://forum.nokia.com, visited April 25, 2005

16. Forum Nokia, "Enhancing the User Experience of Mobile Services with WAP Push", Version 1.1 19 June 2003, http://forum.nokia.com, visited April 25, 2005

17. Rahul Kumar Gupta, "Application Architecture for Disparate Client Support", http://www.devx.com/DevX/Article/9778, visited May 20, 2005

18. "WAP", http://www.m-indya.com/mwap/wap/wap.htm, visited march 23, 2005

19. Qusay H. Mahmoud, "WAP for Java developers, Develop WAP applications with Java servlets and JSP", http://www.javaworld.com/javaworld/jw-06-2000/jw-0602-wap.html, Visited March 2005

20. Eija Kaasinen, Matti Aaltonen, Juha Kolari, Suvi Melakoski, Timo Laakko, "Two Approaches to Bringing Internet Services to WAP Devices", http://www9.org/w9cdrom/228/228.html, visited May 2005

21. Forum Nokia, "Guidelines for Creating Web Content for Mobile and PC Browsing", version 1.0 september 27 2004, http://forum.nokia.com, visited April 25 2005

22. Nokia Corporation, “Browsing on Mobile Devices”, july 2004, www.nokia.com, visited march 2005

23. Federal Cooperation Commission, “Final Requirement Analysis Plan, Cooperative Marketing Information System & Network Study”, Volume I, July 2004

24. Federal Cooperative Commission, “System Design Final Report”, September 2004

25. Ethiopian telecommunication Authorities, http://www.ethionet.et/EthioMobile/index.html, visited may 2005

26. Forum Nokia, “Nokia Mobile Browser 4.0 SDK User’s Guide: Version 4.0”, April 16,2003, www.forum.Nokia.com, visited April 2005

27. Forum Nokia, “Nokia Mobile Internet Toolkit User’s Guide: Version 4.1”, May 2004, www.forum.Nokia.com, visited April 2005

28. Forum Nokia, “Nokia WAP Gateway Simulator User’s Guide: Version 4.0”, May 2003, www.forum.Nokia.com, visited April 2005

29. WAP Forum, “White Paper -Wireless Application Protocol: Wireless Internet Today”, June 2000, www.wapforum.org, visited march 2005

30. Sena mirmir,“Sadiss 1.0, Unicode Text Editor for Ethiopic”, http://www.senamirmir.com/projects/sadiss/sadiss_proj.html, visited May 2005

31. Jonathan Knudsen, “Wireless Development Tutorial Part II”, September 2003, http://developers.sun.com/techtopics/mobility/midp/articles/tutorial2/ Visited on April 21, 2005

32. Openwave Systems Inc, “Openwave WAP Push Library, Developer’s Guide”, Java Edition Release 1, http://www.openwave.com, Visited on April 11, 2005

Declaration

This project has not been presented for a degree in any other university. It is an original work and that all sources of materials used for the project has been duly acknowledged.

Name

Teferi Assefa _____

Signature

Advisor confirmation:

Name:

Solomon Atnafu (PhD) _____

Signature

Department of Computer Science,

Addis Ababa University,

Addis Ababa, Ethiopia

Date of Submission: July 2005