

DICTIONARY BASED MOBILE PHONE APPLICATION FOR PUNJABI LANGUAGE

Thesis submitted in partial fulfillment of the requirements for the
award of degree of

Master of Engineering
in
Computer Science & Engineering

By:
Varinder pal Singh
(80632501)

Under the supervision of:
Parteek Bhatia
Senior Lecturer



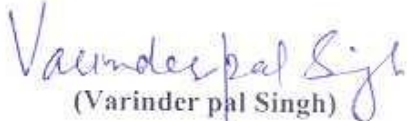
COMPUTER SCIENCE AND ENGINEERING DEPARTMENT
THAPAR UNIVERSITY
PATIALA – 147004

MAY 2009


Certificate

I hereby certify that the work which is being presented in the thesis entitled, **“Dictionary Based Mobile Phone Application For Punjabi Language”**, in partial fulfillment of the requirements for the award of degree of Master of Engineering in Computer Science and Engineering submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of Parteek Bhatia and refers other researcher's works which are duly listed in the reference section.

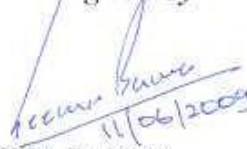
The matter presented in this thesis has not been submitted for the award of any other degree of this or any other university.


(Varinder pal Singh)

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


(Parteek Bhatia)
Computer Science and Engineering Department
Thapar University, Patiala

Countersigned by


(SEEMA BAWA)
Professor & Head
Computer Science & Engineering, Department
Thapar University,
Patiala


(R.K.SHARMA)
Dean (Academic Affairs)
Thapar University,
Patiala.

Acknowledgement

I am extremely thankful to my Guide Mr. Parteek Bhatia Senior Lecturer Computer Science and Engineering of Thapar University for his valuable advice, motivation, guidance, encouragement, moral support, sincere efforts and attitude with which he solved my queries and make this thesis possible. It has being a great pleasure and experience to work under him. I am honored to work with him.

The support from Dr. (Mrs.) Seema Bawa, Professor and Head Computer Science and Engineering Department, Thapar University, Patiala is immense and invaluable. I am highly thankful for her support, guidance and encouragement. I am indebted to her suggestions and help during my work.

I would like to thank Inderveer Kaur (PG Coordinator) Lecturer (Selection Grade) Computer Science and Engineering Department, Thapar University, Patiala and other faculty members for their assistance, suggestions and stimulating decisions during the period of this research.

Finally, I wish to thank my dearest family and friends for all their love, encouragement and constant support without which I could not complete this work. Last but not least, I would like to thank God for not letting me down in time of crisis and always being with me in my good and bad times.

Varinder pal Singh
(80632501)

Abstract

Punjabi people constitute second largest Non Resident Indian population in all around the world from India. Punjabi language is mother tongue of more than 110 million people. The new generation of these NRI people does not have understanding of the Punjabi language. The Punjabi people travel most in the world and they face the problem of understanding the universal language English and NRI Punjabi people do not understand Punjabi Language.

This thesis is an endeavor to provide the Punjabi translation, transliteration and dictionary applications in the hands of people through mobile phones. With the advancements in mobile computing, the mobile phone can be used for connecting with the already existing web based language interpretation software's. The client system on mobile phones can be developed in Java Micro Edition that can be connected to the existing web based software.

J2ME is a very small Java application environment, which is basically a framework for the deployment and use of Java technology in the post-PC world, dominated by mobile and constrained devices (set top box and personal digital assistant). This thesis is about the mobile phone, different types of the existing Punjabi (Gurmukhi) Language interpretation software's and the J2ME programming environment. We have implemented an application on mobile phone using the J2ME environment that translates the English (Roman) to Punjabi (Gurmukhi) language.

Table of contents

CERTIFICATE	I
ACKNOWLEDGEMENT	II
ABSTRACT	III
TABLE OF CONTENTS.....	IV
LIST OF FIGURES	VI
LIST OF TABLES	VII
CHAPTER 1	1
CROSSING LANGUAGE BARRIERS USING MOBILES	1
CHAPTER 2	4
MOBILE PHONE FOR LANGUAGE TRANSLATION	4
2.1 INTRODUCTION TO MOBILE SYSTEM.....	4
2.1.1 <i>Different Types of the mobile systems</i>	4
2.2 EVOLUTION OF THE MOBILE PHONE.....	5
2.2.1 <i>Advanced Mobile Phone System/ First Generation</i>	5
2.2.2 <i>Second Generation</i>	6
2.2.3 <i>Second Generation (2G+) Wireless Networks</i>	7
2.2.4 <i>Third Generation (3G) Wireless Networks</i>	8
2.2.5 <i>Cellular Network</i>	10
2.3 INSIDE A DIGITAL MOBILE PHONE	11
2.4 MOBILE OPERATING SYSTEMS.....	12
2.5 APPLICATIONS OF MOBILE COMPUTING	13
2.5.1 <i>Horizontal Applications</i>	13
2.5.2 <i>Vertical Applications</i>	14
2.6 PROGRAMMING PLATFORMS FOR MOBILE COMPUTING.....	15
2.6.1 <i>J2ME vs BREW</i>	17
2.6.2 <i>Java ME on Symbian Operating System</i>	18
2.7 JAVA IN MOBILE PHONES	19
2.7.1 <i>Introduction to J2ME</i>	20
2.7.2 <i>Architecture of J2ME</i>	20
2.7.3 <i>Mobile Information Device Profile (MIDP) Layered Architecture</i>	23
2.7.4 <i>Characteristics of MIDP</i>	24
2.7.5 <i>Programming Architecture of MIDP 2.0</i>	25
2.7.6 <i>MIDlet</i>	27
2.7.7 <i>Event handling in J2ME</i>	28
2.7.8 <i>User Interface Design in MIDP</i>	28
2.6.9 <i>CLDC API</i>	30
2.7.10 <i>MIDP API</i>	34
2.8 HOW JAVA APPLICATIONS RUN ON MOBILE PHONES.....	35

2.9 J2ME WEB SERVICES API (WSA).....	35
2.10 LANGUAGE TRANSLITERATION	36
2.10.1 English to Punjabi Dictionary	38
2.10.2 Techniques of Language Translation.....	38
CHAPTER 3	40
PROBLEM STATEMENT.....	40
3.1 GAP ANALYSIS.....	40
3.2 METHODOLOGY	41
3.3 HOW IT HELP?	41
CHAPTER 4	43
DESIGN AND IMPLEMENTATION	43
4.1 FEATURES OF NETBEANS.....	43
4.1.1 The Wireless Toolkit Features	45
4.1.2 Installation and Configuring of Netbeans 6.0 with J2ME.....	46
4.1.3 Creating the MIDP/CLDC Application.....	47
4.2 APPLICATION DEVELOPMENT PROCESS.....	48
4.2.1 Work Flow Diagram of the application	49
CHAPTER 5	52
RESULTS.....	52
5.1 CASE 1.....	53
5.2 CASE 2.....	54
5.3 CASE 3.....	55
CHAPTER 6	56
CONCLUSIONS AND FUTURE SCOPE	56
6.1 CONCLUSION	56
6.2 FUTURE SCOPE.....	57
REFERENCES	59
LIST OF PUBLICATION	62
APPENDIX	63

LIST OF FIGURES

Figure 2.1: Hierarchy of J2ME.....	15
Figure 2.2: Conceptual View of BREW.....	16
Figure 2.3 Architecture of J2ME.....	20
Figure 2.4: MIDP Architecture.....	24
Figure 2.5: Programming Architecture of J2ME.....	27
Figure 2.6: WSA working.....	36
Figure 4.1: MIDlet Lifecycle.....	48
Figure 4.2: Netbean Mobility Application.....	49
Figure 4.3: Workflow Diagram of application.....	50
Figure 5.1 : Starting Screen of Application.....	52
Figure 5.2: Application used as Punjabi Dictionary.....	53
Figure 5.3: Snapshot of Application used as Punjabi Translator.....	54
Figure 5.4: Word not Found Condition.....	55

LIST OF TABLES

Table 2.1: Comparison among 2G, 2G+ and 3G Wireless Networks.....9
Table 2.2: Different Mobile Profile.....22
Table 2.3: Features of Java Standards.....23

Chapter 1

Crossing Language Barriers using Mobiles

The communication among human beings is done by language. This world is collection of diverse cultures and societies. We use different types of languages to communicate among each other. The food, trade, job brings the different societies, culture closer and there the need of language understanding arises. The translators of Languages are used to overcome this inter state/ societies communications barriers.

With the advancement in the technology the information flows at the speed of the light from each corner of the world to another. People use to travel, view and communicate in different parts of the globe at very fast pace. The world has become a village with new technologies like computer, internet and telecommunication. There is a need to bridge the language barriers that exist all around the world. We can use the technology for crossing these language barriers. The computers can be used to translate the source language to desired target language with only pressing of few keys.

Also, the advent of the mobile phone has transformed the way of communication between the human beings. The mobile phone becomes the important part of human life and its growth is astoundingly exponential. Mobile phones are the integral part of our life. The SMS (Short Message Service) [26] of the mobile phone becomes the habit of our society.

The robust and reliable wireless communication infrastructure has given cellular phone developer enough space to enhance the features of their products. The 2G and 3G wireless technologies have increased the data transfer, voice quality of the mobile phones. The improvement in mobile hardware and increase in its memory has helped the developer to embed their software's like operating systems, application software's on the device. That gives an opportunity for programmer to flex their muscles on the mobile or constrained devices

Machine aided translation system [14] available on web may certainly be proved to be more useful if become available on mobile phone with a different interface to provide interpersonal, domain specific communication across language barrier. Machine translation is used to translate from one natural language to another natural language. The different techniques used to do translation are rule based, statistical based and example based. The machine translation is used on computer systems that have fast data processing capabilities.

The Punjabi people are adventurous and constitute one of the biggest Non Resident Indian populations of the country India. The mobile density in Punjab is highest in India. This intelligent device can also be used to help the rural as well as the NRI in their language understanding problems. We have tried to use the mobile phones for the language understanding among the people of this region. Mobile phones are resource scarce entities while, the machine language translation is resource hungry process. The study of the mobile system suggests that we can use the latter process by SMS or MIDlets [3].

The language translation is very subjective. The software cannot be directly used on the limited resource system like mobile phones. But these services can be utilized with the help of Web Services, SMS and MIDlets on mobile systems. Java is the platform on which the MIDlets are developed for specific domains. The J2ME [29] is used as base layer for development of application in java on mobile phones. The java application uses Mobile Information Device Profile (MIDP) [18] to develop an application for mobile phone. In this thesis we have created a simple MIDlet *i.e* an application on MIDP for mobile phones. This application converts the English (Roman) text into the Punjabi (Gurmikhi) text using direct translation techniques.

Short message service is the crisp and the popular way to share information among the masses. But their complex and costly implementation at the service provider side is the bone of contention for the developer. The use of MIDlets for mobile is an option for designing and developing an application at the mobile phone only. Also the Web Services [7] can be used to attach the existing translation systems with the mobile systems.

We have implemented a mobile based application with the help of J2ME programming environment. This application uses the MIDP profile by J2ME for Mobile phones and sun wireless toolkit [6] of Netbean [17] Integrated Development Environment. The user will enter the English text as input in emulator screen and get the translated Punjabi text as output.

Further the web application can be developed that can be used to connect the mobile with the web service. That can be used to connect the existing website that implements machine translation or transliteration.

Mobile Phone for Language Translation

2.1 Introduction to Mobile system

The mobile systems are like mini computers that are used by us in our day to day life style. The mobile phones are in between the extreme laptop and display only (like info pad) mobile systems. Once thought of as luxury, cell phones are now often considered a necessity and a lifeline in time of an emergency. Mobile phones are not only used to transfer voice but also used to transmit the data.

2.1.1 Different Types of the mobile systems

The Electronic gadgets with less weight, computational facilities, communication features are known as mobile systems. They are distinguished based on the said features.

The following are different types of mobile systems

- Laptop: It is a personal computer with mobility. It is used for high computational programs. The communication on this device is specific to data. The voice communication is rarely used on this type of the machine. The mobility is still restricted because of its size and the weight. The high costs, less mobility, nearly zero voice communication are demerits of these type of the devices. High computational power, fast data transferring features are a few merits of these devices.
- Personal Digital Assistant (PDA): Digital device with high mobility and processing features. These mobile systems are used for Email, data processing and the other office management facilities. The emphasis on these types of the phones is voice communication along with the rich computational processing facilities. High cost and vendor specific features limit the scope of application development on these types of the devices.
- Mobile Phones: A mobile phone/cell phone is an electronic device used for voice or data communication over a network of specialized base stations

known as cell sites. It acts as the sophisticated radio with which you can talk to anyone on the planet from anywhere. This technology is the essence of present world and main focus of our study. The mobile phones are categorized as follows:

- **Featured Mobile Phone:** The phones with additional features namely data processing capabilities, email, power point presentation, camera, video recording, memory capacity more than 8GB are called featured phones. These phones are proprietary items like Window Mobile, Blackberry, iPhone *etc.*
- **Simple Mobile Phone:** The mobile phone with limited resources and features are named as simple phones. These phones are cheap and more robust as compare to featured phones. Although these phone are now matching in features with the featured phones.

2.2 Evolution of the Mobile Phone

The half-duplex communication is the starting point for the radio (sender/receiver) communications. The half duplex devices were used at vehicles like taxi/buses and they were not connected to the telephones. It was in 1940 when USA used the Walkie-Talkie and later developed a large hand-held two-way radio for the US military. The connectivity, speed, reliability and mobility were the bottleneck with these systems.

Radio telephony was introduced on a large scale in German tanks during the Second World War. The first fully automatic mobile phone system, called MTA (Mobile Telephone system A), was developed by Ericsson and commercially released in Sweden in 1956. One of the first successful public commercial mobile phone networks was the ARP (Autoradiopuhelin, "car radio phone network") [9] in Finland, launched in 1971. Posthumously, ARP is sometimes viewed as a zero generation (0G) cellular network

2.2.1 Advanced Mobile Phone System/ First Generation

The standard was developed by Bell labs and officially introduced in America in 1983. It was the Analog mobile phone standard. AMPS was the first generation

cellular technology that uses separate frequencies. In AMPS [9], the cell centers can flexibly assign channels to handsets based on signal strength, allowing the same frequency to be re-used in various locations without interference. This allowed a larger number of phones to be supported over a geographical area.

AMPS pioneers fathered the term "cellular" because of its use of small hexagonal "cells" within a system. AMPS cellular service operates in the 800 MHz Cellular FM band. Each carrier within a market uses a specified "block" of frequencies consisting of 21 control channels and 395 voice channels. The Motorola was the first company that produced handheld mobile set in US. Since it is an analog standard, it is very susceptible to static and noise and has no protection from eavesdropping using a scanner.

2.2.2 Second Generation

The second generation (2G) [9] of the wireless mobile network was based on low-band digital data signaling. The most popular 2G wireless technology is known as Global Systems for Mobile Communications (GSM) [1]. GSM systems, first implemented in 1991, are now operating in about 140 countries and territories around the world.

GSM technology is a combination of Frequency Division Multiple Access (FDMA) and Time Division Multiple Access (TDMA). The first GSM systems used a 25MHz frequency spectrum in the 900MHz band. FDMA is used to divide the available 25MHz of bandwidth into 124 carrier frequencies of 200 kHz each. Each frequency is then divided using a TDMA scheme into eight timeslots. The use of separate timeslots for transmission and reception simplifies the electronics in the mobile units. While GSM and other TDMA-based systems have become the dominant 2G wireless technologies, CDMA technology is recognized as providing clearer voice quality with less background noise, fewer dropped calls, enhanced security, greater reliability and greater network capacity. 2G wireless technology can handle some data capabilities such as fax and short message service at the data rate of up to 9.6 kbps, but it is not suitable for web browsing and multimedia applications

2.2.3 Second Generation (2G+) Wireless Networks

The effective data rate of 2G circuit-switched wireless systems is relatively slow. As a result, GSM, PDC and other TDMA-based mobile system providers and carriers have developed 2G+ [9] technology that is packet-based and increases the data communication speeds to as high as 384kbps. The emerged technologies in 2G+ are

- High Speed Circuit Switched Data (HSCSD) is one step towards 3G [1] wideband mobile data networks. This circuit-switched technology improves the data rates up to 57.6kbps by introducing 14.4 kbps data coding and by aggregating 4 radio channels timeslots of 14.4 kbps.
- General Packet Radio Service (GPRS) is an intermediate step that is designed to allow the GSM world to implement a full range of internet services without waiting for the deployment of full-scale 3G wireless systems. GPRS technology is packet-based and designed to work in parallel with the 2G GSM, PDC and TDMA systems that are used for voice communications and for table look-up to obtain GPRS [1] user profiles in the Location Register databases. GPRS uses a multiple of the 1 to 8 radio channel timeslots in the 200kHz-frequency band allocated for a carrier frequency to enable data speeds of up to 115kbps. The data is packetized and transported over Public Land Mobile Networks (PLMN) using an IP backbone so that mobile users can access services on the Internet, such as SMTP/POP-based e-mail, ftp and HTTP-based Web services.
- Enhanced Data rate for Global Evolution (EDGE) [1] technology is a standard that has been specified to enhance the throughput per timeslot for both HSCSD and GPRS. The enhancement of HSCSD is called ECSD, whereas the enhancement of GPRS is called EGPRS. In ECSD, the maximum data rate will not increase from 64 kbps due to the restrictions in the A interface, but the data rate per timeslot will triple. Similarly, in EGPRS, the data rate per timeslot will triple and the peak throughput, including all eight timeslots in the radio interface, will exceed 384 kbps.

2.2.4 Third Generation (3G) Wireless Networks

3G wireless technology represents the convergence of various 2G wireless telecommunications systems into a single global system that includes both terrestrial and satellite components. One of the most important aspects of 3G wireless technologies is its ability to unify existing cellular standards, such as CDMA [9], GSM, and TDMA, under one umbrella. The following three air interface modes accomplish this result: wideband CDMA, CDMA2000 and the Universal Wireless Communication (UWC-136) interfaces.

W-CDMA will require bandwidth of between 5 MHz and 10 MHz, making it a suitable platform for higher capacity applications. It can be overlaid onto existing GSM, TDMA (IS-36) and IS95 networks. W-CDMA [1] networks will be used for high-capacity applications and 2G digital wireless systems will be used for voice calls

The second radio interface is CDMA2000 which is backward compatible with the second generation CDMA IS-95 standard predominantly used in US. The third radio interface, Universal Wireless Communications – UWC-136, also called IS-136HS, was proposed by the TIA and designed to comply with ANSI-136, the North American TDMA standard.

3G wireless networks consist of a Radio Access Network (RAN) and a core network. The core network consists of a packet-switched domain, which includes 3G SGSNs and GGSNs, which provide the same functionality that they provide in a GPRS system, and a circuit-switched domain, which includes 3G MSC for switching of voice calls. Charging for services and access is done through the Charging Gateway Function (CGF), which is also part of the core network. RAN functionality is independent from the core network functionality. The access network provides a core network technology independent access for mobile terminals to different types of core networks and network services. Either core network domain can access any appropriate RAN service; e.g. it should be possible to access a “speech” radio access bearer from the packet switched domain.

Features	2G	2G+	3G
Core Network	TDM transport	TDM, Frame Relay transport	ATM, IP transport
Handsets	Voice only terminals	Dual mode TDMA and CDMA Voice and data terminals WAP no multimedia support	New type of terminal Multiple modes Voice, data and video terminals WAP, multimedia mgmt
Data Rates	Up to 9.6 Kbps	Up to 57.6 Kbps (HSCSD) Up to 115Kbps (GPRS) Up to 384 Kbps (EDGE)	Up to 2Mbps
Applications	Advanced voice, Short Message Service (SMS)	SMS, Internet	Internet, multimedia
Roaming	Restricted, not global	Restricted, not global	Global
Radio Access	FDMA, TDMA, CDMA	TDMA, CDMA, EDGE	W-CDMA, CDMA2000, IWC-136

Table 2.1: Comparison among 2G, 2G+ and 3G Wireless Networks [1].

The network used by mobile for communication incepted with analog communication also called zero generation and matured with digital communication that is second generation communication. The spectrum used in present scenario is called second generation (2G+) communication. The mobile phones with processing capabilities are used for communication and data transfer. The current generation phones provide the programmer a space for application development. The operation system for the mobile systems like Windows Mobile, Symbian provide a full featured mobile phone in the hands of the user. Phone with capabilities of processing data give a developer a thought to develop applications like share market tickers, railway ticket status, maps, 2D games etc.

The use of assembly language prohibited the developer jump into mobile programming. The improved version of mobile phones uses the different programming languages like BREW, C/C++, Open C and the most popular language JAVA.

2.2.5 Cellular Network

The name itself described the structure of the network. The cellular radio transmission is the base of all the wireless networks. Before the advent of cellular technology, capacity was enhanced through a division of frequencies, and the resulting addition of available channels. However, this reduced the total bandwidth available to each user, affecting the quality of service. Cellular technology allowed for the division of geographical areas, rather than frequencies, leading to a more efficient use of the radio spectrum.

In a cellular network, cells are generally organized in groups of seven to form a cluster. There is a “cell site” or “base station” at the centre of each cell, which houses the transmitter/receiver antennae and switching equipment. The size of a cell depends on the density of subscribers in an area: for instance, in a densely populated area, the capacity of the network can be improved by reducing the size of a cell or by adding more overlapping cells. This increases the number of channels available without increasing the actual number of frequencies being used. All base stations of each cell are connected to a central point, called the Mobile Switching Office (MSO), either by fixed lines or microwave. The MSO is generally connected to the PSTN (Public Switched Telephone Network). Access Technologies are described below:-

- **FDMA:** Frequency Division Multiple Access (FDMA) is the most common analog system. It is a technique whereby spectrum is divided up into frequencies and then assigned to users. With FDMA, only one subscriber at any given time is assigned to a channel. The channel therefore is closed to other conversations until the initial call is finished, or until it is handed-off to a different channel. A “full-duplex” FDMA [1] transmission requires two

channels, one for transmitting and the other for receiving. FDMA has been used for first generation analog systems.

- TDMA: Time Division Multiple Access (TDMA) improves spectrum capacity by splitting each frequency into time slots. TDMA allows each user to access the entire radio frequency channel for the short period of a call. Other users share this same frequency channel at different time slots. The base station continually switches from user to user on the channel. TDMA is the dominant technology for the second generation mobile cellular networks.
- CDMA: Code Division Multiple Access is based on “spread” spectrum technology. Since it is suitable for encrypted transmissions, it has long been used for military purposes. CDMA [1] increases spectrum capacity by allowing all users to occupy all channels at the same time. Transmissions are spread over the whole radio band, and each voice or data call are assigned a unique code to differentiate from the other calls carried over the same spectrum. CDMA allows for a “soft hand-off”, which means that terminals can communicate with several base stations at the same time. The dominant radio interface for third-generation mobile, or IMT-2000, will be a wideband version of CDMA with three modes (IMT-DS, IMT-MC and IMT-TC).

2.3 Inside a Digital Mobile Phone

A mobile phone [10] typically consists of a single circuit board that is also known as heart of phone. The different functional areas of a mobile phone are as follows:

- Radio frequency – receiver and transmitter: The radio frequency (RF) and power section handles power management and recharging, and also deals with the hundreds of FM channels. The RF amplifiers handle signals traveling to and from the antenna
- Digital signal processing: The digital signal processor (DSP) is a highly customized processor designed to perform signal-manipulation calculations at high speed.

- Analogue/Digital conversion: The analog-to-digital and digital-to-analog conversion chips translate the outgoing audio signal from analog to digital and the incoming signal from digital back to analog.
- Control processor: The microprocessor handles all of the housekeeping chores for the keyboard and display, deals with command and control signaling with the base station and also coordinates the rest of the functions on the board.
- Battery: The power supplier for the mobile that provide long backup without charging frequently.
- Memory: The ROM and Flash memory chips provide storage for the phone's operating system and customizable features, such as the phone directory

In addition to these elements a Subscriber Identity Module (SIM) or for 3G a Universal Subscriber Identity Module (USIM) is included.

2.4 Mobile Operating Systems

The controller of smart device that provide developer the base on which the applications can be developed and deployed for the user is called mobile operating system. Different types of mobile operating systems are as follows:

- Symbian: The mobile phone device operating system with associated libraries, user interface, frameworks and reference implementations of common tools, developed by Symbian Ltd. with associated libraries, user interface, frameworks and reference implementations of common tools, developed by Symbian Ltd. Symbian OS programming is event-based, and the CPU is switched off when applications are not directly dealing with an event. This is achieved through a programming idiom called active objects. Similarly the OS approach to threads vs. processes is driven by reducing overheads.
- Windows Mobile: The operating system based on Microsoft WIN32 bit API. Windows Mobile include Pocket PCs, Smartphones, Portable Media Centers, and on-board computers for certain automobiles. It is designed to be somewhat similar to desktop versions of Windows, feature-wise and aesthetically. Microsoft typically releases Windows Mobile Software

development kits (SDKs) that work in conjunction with their Visual Studio development environment.

- RIM: Research In Motion (RIM) is a leading designer, manufacturer and marketer of innovative wireless solutions for the worldwide mobile communications market. RIM provides platforms and solutions for seamless access to time-sensitive information including email, phone, text messaging (SMS and MMS), Internet and intranet-based applications.
- MAC OS iPhone: The mobile phone operating system developed by Apple Corporation for iPhone touch. iPhone OS has four abstraction layers: the Core OS layer, the Core Services layer, the Media layer, and the Cocoa Touch layer. The operating system takes less than 240 Megabytes of the device's total memory storage.
- Palm OS: Palm OS is designed for ease of use with a touchscreen-based graphical user interface. It is provided with a suite of basic applications for personal information management. The C/C++ programming language is used to develop the application on this operating system.

2.5 Applications of Mobile Computing

Mobile Computing allows transmission of data, via a computer, without connecting to a fixed physical link. It requires the wireless network support and handoff the network change during its mobility. A mature mobile computing is the base for Ubiquitous computing [4].

Although many varied applications exist, mobile computing applications can generally be divided into two categories--horizontal and vertical.

2.5.1 Horizontal Applications

Horizontal applications have broad-based appeal and include software that performs functions such as: (a) email; (b) Web browsing; (c) word processing; (d) scheduling; (e) contact management; (f) to-do lists; (g) messaging; (h) presentation. These types of applications usually come standard on Palmtops,

Clamshells, and laptops with systems software such as Microsoft Windows operating system.

2.5.2 Vertical Applications

Vertical applications are industry-specific and only have appeal within the specific industry for which the application was written. These vertical applications are often transaction oriented and normally interface with a corporate database. A few of them are as follows:

- **Emergency Services:** They can be used to connect with hospital system and can retrieve the valuable information like First Aid etc.
- **For Estate Agents:** Estate agents can work either at home or out in the field. With mobile computers they can be more productive. They can obtain current real estate information by accessing multiple listing services, which they can do from home, office or car when out with clients
- **In courts:** Defense counsels can take mobile computers in court. When the opposing counsel references a case which they are not familiar, they can use the computer to get direct, real-time access to on-line legal database services, where they can gather information on the case and related precedents
- **In companies:** Managers can use mobile computers in, say, and critical presentations to major customers. They can communicate with the office about possible new offers and call meetings for discussing responds to the new proposals
- **Stock Information Collation/Control:** the information from the web servers at stock exchange can be connected to the hand held devices through software like MIDlets (small code in java run on mobile phones).
- **Mobile Automation:** General business travelers also reap the benefits of mobile computing. E-mail, spreadsheets, presentations, and word processing are the four primary tasks accomplished by these business travelers. Laptops, Palmtops, and portable Clamshell computers with usable-size keyboards enable businesspeople to stay in touch and accomplish the tasks they need for job effectiveness.

2.6 Programming Platforms for Mobile Computing

A few technologies that are currently used by the developers for developing mobile applications are listed below

- J2ME (Java 2 Micro-Edition)

Sun Microsystems defines J2ME [29] as "a highly optimized Java run-time environment targeting a wide range of consumer products, including pagers, cellular phones, screen-phones, digital set-top boxes and car navigation systems." By adding a virtual machine between applications and operating systems, J2ME programs are inherently platform independent and thus increase productivity for the programs running on mobile devices. The virtual machine of J2ME [8], Kilobyte Virtual Machine (KVM), is much smaller than traditional Java Virtual Machine (JVM). It is designed to run under the constraints of the limited resources available on micro devices. Currently, J2ME has been one of the most widely-adopted developing platforms for mobile devices. The conceptual view of J2ME is shown in Figure 2.1

Java 2 Micro-Edition Connected Limited Device Configuration (J2ME CLDC) is the platform of choice when it comes to running mobile applications on resource constrained devices (cell phones, set-top boxes, etc.). The Mobile Information Device Profile (MIDP) with the Connected Limited Device Configuration (CLDC) is the Java runtime environment for today's mobile information devices.

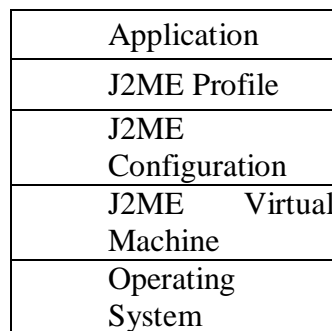


Figure 2.1: Hierarchy of J2ME

- BREW (Binary Runtime Environment for Wireless)

Binary Runtime Environment for Wireless, An application development environment from Qualcomm for enhanced cellphone services (e-mail, games).

BREW is a software platform that can download and run small programs for

playing games, sending messages, sharing photos, etc. The main advantage of BREW [23] platforms is that the application developers can easily port their applications between all Qualcomm devices. BREW acts between the application and the wireless device on-chip operating system in order to allow programmers to develop applications without needing to code for system interface or understand wireless applications. BREW is a thin client (about 150K) that sits between a software application and the Application Specific Integrated Circuit (ASIC) level software. Thus developers can write to BREW without knowing or caring about the device's chipset or air interface. While it's true that CDMA is Qualcomm's specialty, BREW is equally capable of running on devices that employ other air interface standards. Figure 2.2 shows the conceptual layering of software on wireless device.

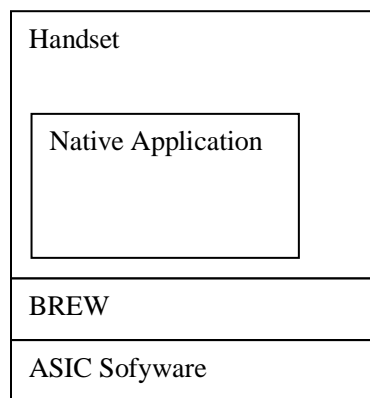


Figure 2.2: Conceptual View of BREW

- C/C++

These are also said as native languages for mobile programming. They directly write in ARM architecture which constitutes the 75 % of current mobile phone market. They provide better usability and more fine grain control over the mobile phones. C++ Symbian [20] has very less features and is tough nut to crack

- Python

It is also known as the scripting language. Ease of development is the characteristic of this language. The cross platform and open source had made this language popular among developers. The standard library, access to full phone functionality and reasonable memory footprint are a few merits of the Python [16] on mobile.

- Flash Lite

It is very productive environment for highly graphic compelling user interfaces and full end to end services. Flash video support for the Google and YouTube services. Flash Lite supports a larger set of ActionScript functionality and Flash 8, requiring a larger memory footprint and more processing power. Flash Lite is supported in the Brew MP SDK and on devices that have licensed the Flash Lite player.

- Widget

It is core web page designed for the specific function and it can be designed with the known languages like HTML, CSS etc. the development time required is just few days.

2.6.1 J2ME vs BREW

All in all, BREW doesn't sound much different from Java. To a user, it presents a similar kind of application experience and to a developer, a more or less similar development model. However, some distinctions between the two are given below.

1. The J2ME philosophy is essentially the Java philosophy, which is to enable developers to write applications that can run unaltered on different devices. J2ME therefore targets general consumer and embedded devices. BREW, on the other hand, targets wireless devices exclusively (specifically CDMA phones). As a result, it's more versatile when it comes to wireless phones—and by the same token, lacks portability.
2. J2ME doesn't consist of a specific distribution system corresponding to the BREW BDS. This may be an advantage as well as a shortcoming. Although it enables free distribution of Java applications, some developers may find the managed revenue distribution system that BREW provides more attractive.
3. BREW currently lacks the extent of development tools that Java developers have access to.
4. Since J2ME is an interpreted language, it should be slower (at least theoretically) than BREW, which is closer to the processor.

5. Java garbage collection can be very effective when it comes to memory issues. These issues can sometime be a serious problem with BREW, with its severe memory limitations.

2.6.2 Java ME on Symbian Operating System

Unlike other operating systems, Symbian has a very rich Java environment, meaning more capabilities than on most feature phones. The Java ME implementation shipped with Symbian OS is best-of-breed, so MIDlets will generally execute at quite acceptable speeds on current smartphones.

Symbian OS [24] still holds the following advantages compared to feature phones when it comes to the management of Java ME applications:

- **Robustness:** Process separation is there for a reason; without it, if a game MIDlet triggers a native bug in the implementation of a JSR library, it can take out the VM and require it to be restarted. On a feature phone this will cause all running MIDlets to die without notification, potentially losing user data. On Symbian OS, MIDlets are fully protected from one another.
- **Platform integration:** S60 and UIQ are already full multitasking environments with facilities for managing an arbitrary number of running processes. On Symbian OS, a user does not have to learn anything new to manage MIDlet multitasking; MIDlets are handled in exactly the same way as native applications. On feature phones, MIDlet multitasking often doesn't sit well with the native UI. Instead it is common for there to be a separate "Java" application with its own task management UI.
- **No arbitrary limitations:** Feature phones typically have to divide up the memory assigned to various features at device creation time, and the memory assigned to Java is fixed. On Symbian OS, because MIDlets are process-based and managed like native applications, they co-operate with the system-wide memory management policy. This allows complex and resource-intensive MIDlets to be run on Symbian OS using the full capabilities of the device without hitting artificially imposed limits.

- Future proofing: although feature phones can now just about handle CLDC/MIDP applications via enhanced VMs, they still cannot handle CDC-based Java platforms like the forthcoming JSR 249: Mobile Service Architecture Advanced and JSR 232: Mobile Operational Management, and there is little prospect of them ever doing so. In contrast, CDC/Foundation is already shipping on a number of Symbian phones.
- Architecture: on Symbian OS, Java ME applications are treated as first-class citizens. MIDlets completely integrate into APPARC and the view server, so switching between MIDlets and native applications is identical.
- Fragmentation: as already discussed, the available set of JSRs is well defined and highly optimized. This largely alleviates fragmentation problems that plague other platforms.

2.7 Java in Mobile Phones

Java is widely used in mobile phones. Over 250 million Java technology-enabled wireless devices from 31 manufacturers are deployed in over 75 carrier networks and half a billion Java Card secure digital identity deployments in smart card and mobile phone environments. This seems to be just the beginning for this platform agnostic software.

The biggest drivers of Java as a suitable technology for mobile phones have been:

1. The need for richer devices like smart phones with advanced features and application.
2. The need for platform independent technology in a multiplatform ridden mobile world.
3. The need for creating customizable devices that can generate brand identity for operators and enable over the air downloading of applications which provide greater Average Revenue Per User (ARPU).

Java [8] seems to hold the promise of simpler cell phone programming, richer games, and more money for cell phone service providers, software that connects to Internet and other services.

2.7.1 Introduction to J2ME

The Sun technologies define the Java 2 Platform, Micro Edition (J2ME™) specification for application development on small, battery-powered devices with limited wireless connections to the Internet. The J2ME specification [29] covers a wide range of devices, from pagers and mobile telephones through set-top boxes and car navigation systems. The J2ME world is divided into configurations and profiles, specifications that describe a Java environment for a specific class of device.

2.7.2 Architecture of J2ME

The J2ME architecture is based on families and categories of devices. A family of devices is made up of a group of categories that have similar requirements for memory and processing power. Together, cellular phones, simple pagers, and simple personal organizers make up a single family of small-footprint devices. In order to support the kind of flexibility and customizable deployment demanded by the family of resource-constrained devices, the J2ME [13] architecture is designed to be modular and scalable. This modularity and scalability is defined by J2ME technology in a complete application runtime model, with four layers of software built upon the host operating system of the device. The Layered architecture of the J2ME is shown in the Figure 2.3.

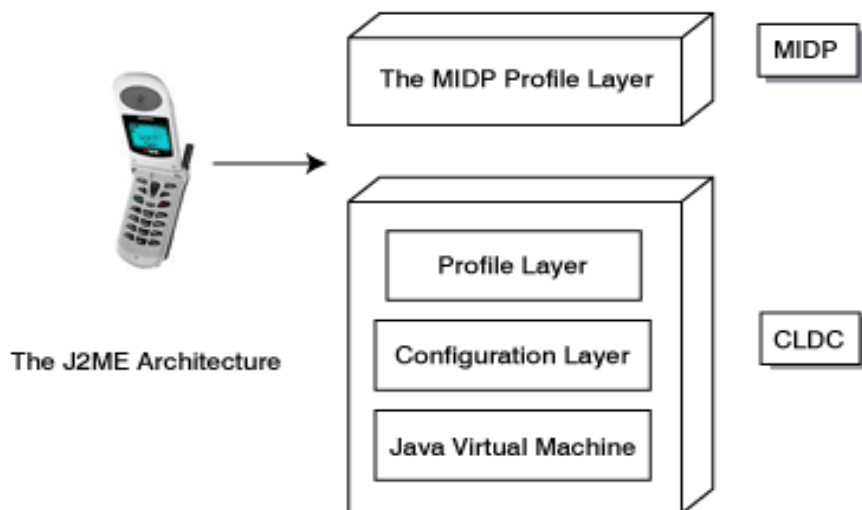


Figure 2.3 Architecture of J2ME [25]

- Java Virtual Machine layer

This layer is an implementation of a Java Virtual Machine that is customized for a particular device's host operating system and supports a particular J2ME configuration. The JVM is of different size and feature with different type of the configurations it has to support. The KVM (K Virtual Machine) is designed for small-memory, resource-constrained, network-connected devices. The CDC contains the C Virtual Machine (CVM). This virtual machine works in 2MB memory and 32 bit processing speed.

- The Configuration Layer:

The fundamental branches of J2ME are configurations. A configuration is a specification that describes a Java Virtual Machine and some set of APIs that are targeted at a specific class of device. In a way, a configuration defines the commonality of the Java platform features and libraries that developers can assume to be available on all devices belonging to a particular category. This layer is less visible to users, but is very important to profile implementers

Configurations are composed of a virtual machine (KVM) and a minimal set of class libraries. They provide the base functionality for a particular range of devices that share similar characteristics, such as network connectivity and memory footprint. The idea is that an application developed for a particular configuration should run on devices supporting that configuration. Currently, there are two J2ME configurations: the Connected Limited Device Configuration (CLDC) [27], and the Connected Device Configuration (CDC). Aimed at the Connected Limited Device Configuration, the KVM is only 40K of object code and needs only “a few tens of kilobytes” at runtime.

The CLDC [27] is the smaller of the two configurations. It is designed for devices with intermittent network connections, slow processors and limited memory devices such as mobile phones, two-way pagers and PDA's. This configuration includes some new classes, not drawn from the J2SE APIs, designed specifically to fit the needs of small-footprint devices. Some of the classes of J2SE are also omitted (for example the `java.applet.Applet` class is dropped and instead the `javax.microedition.midlet.MIDlet` class is added in CLDC). Currently there are 2

versions of CLDC, 1.0 & 1.1. The CLDC specification version 1.1 is an incremental release that is intended to be fully backwards compatible with CLDC specification version 1.0. Important new functionality, such as floating point support, has been added.

- Profile layer

A profile is a set of standard APIs that support a narrower device category within the configuration framework. In order to provide a runtime environment targeted at specific device categories, configurations must be combined with profiles, which are a collection of APIs that further define the application lifecycle model, the user interface, and access to device specific properties. Hence it is clear that profiles are specific to configurations *i.e.* CLDC & CDC have different profiles as shown below.

CLDC Profile:	Mobile Information Device Profile (MIDP)
CDC Profile:	Foundation Profile (FP) (Compatible with CLDC), Personal Basis Profile (PBP) & Personal Profile (PP)

Table 2.2: Different Mobile Profile

- MIDP layer:

The Mobile Information Device Profile (MIDP) [30] is a set of Java APIs that addresses issues such as user interface, persistence storage, and networking. The Java Virtual Machine layer, configuration layer, and profile layer together constitute the Connected Limited Device Configuration (CLDC). The MID Profile and CLDC provide a standard runtime environment that allows new applications and services to be dynamically deployed on end-user devices.

- Optional Packages

The J2ME platform can be further enhanced by combining various optional packages with CLDC/CDC, and their corresponding profiles. These packages are created to address very specific market requirements. They offer standard APIs for using both existing and emerging technologies such as:

- Bluetooth
- Web services

- Wireless messaging
- Multimedia
- Database connectivity

Difference between the J2EE, J2SE and J2ME

Java had revolutionized the internet programming. It's a true object oriented language. The main feature is portability of the language *i.e.* programs once compiled can run on any machine irrespective of operating system and computer architecture of machine. But this increases the size of the programming language. Java Micro edition was introduced that use less number of bytes in memory and become popular with mobile/embedded systems. The Table 2.3 briefly shows the prominent difference among java standards

S. No.	Features	J2EE	J2SE	J2ME
1	Core Java Language	JVM	JVM	CVM/KVM
2	Platform	Server, Workstation	PC, Notepad, Laptops	Pagers, Cellular phones, Screen-Phones, Digital set-top boxes etc
3	Memory Requirements	Upto 64MB	64MB to 1MB	160 KB to 512 KB
4	Processor specifications	64 bit	32 bit	16 bit

Table 2.3: Features of Java Standards

2.7.3 Mobile Information Device Profile (MIDP) Layered Architecture

The Mobile Information Device Profile (MIDP) is designed for mobile phones and entry-level PDA's. It offers the core application functionality required by mobile applications, including the user interface, network connectivity, local data storage and application management. Combined with CLDC, MIDP provides a Java runtime environment that is suitable to the capabilities of handheld devices. There are three versions for this: version 1.0, 2.0 and 2.1. This profile layer resides on the CLDC layer. The CLDC layer resides on native operating system. The Conceptual Architecture of MIDP is shown in Figure 2.4. The application can

also be developed by Original Equipment Manufacturer. The programmer can use J2ME to develop desired application of his need.

MIDP App	OEM App	Native App
MIDP	OEM classes	
CLDC		
Native System		
Mobile Information		

Figure 2.4: MIDP Architecture

2.7.4 Characteristics of MIDP

MIDP enables truly networked applications on mobile information devices. To download an MIDP application, the user browses a list of applications stored on a Web server. MIDP [5] applications provide the foundation for highly graphical and intuitive applications. Its graphical user interface is optimized for the small display size, input methods, and other native features of mobile devices.

- Mobile user interface: MIDP [18] features an application programming interface (API) that shields developers from the complexity of building portable applications. This high-level API enables developers to build easy-to-use, highly graphical, and portable applications optimized for mobile information devices, and reduces development effort Multimedia and game functionality. User interface functionality includes predefined screens for displaying and selecting lists, editing text, popping up alert dialogs, and adding scrolling tickers. All screens and items are device-aware, with built-in support for native display size, input, and navigation capabilities. They enable developers to define highly portable, flexible user interfaces that change layout and navigation to fully leverage each device.

- **Multimedia and game functionality:** The game API adds game-specific functions, such as sprites and tiled layers that take advantage of native device graphic capabilities. Built-in audio provides support for tones, tone sequences, and Windows Wave (WAV) formatted files. In addition, developers can use the Mobile Media API (MMAPI), an optional package for MIDP, to add video and other rich multimedia content.
- **Extensive connectivity:** It supports leading connectivity standards, including HTTP, HTTPS, datagram, sockets, server sockets, and serial port communication. MIDP also supports the Short Message Service (SMS) and Cell Broadcast Service (CBS) capabilities of Global System for Mobile Communications (GSM) and CDMA networks, through the Wireless Messaging API (WMA) optional package. MIDP also supports a server push model. A push registry keeps track of applications registered to receive inbound information from the network. When information arrives, the device decides whether to start the application based on user preferences. This push architecture allows developers to include alerts, messaging, and broadcasts in MIDP applications, as well as leverage the event-driven capabilities of devices and carrier networks
- **Over-the-air provisioning:** A major benefit of MIDP [11] is its ability to dynamically deploy and update applications over the air (OTA). The MIDP specification defines how MIDP applications are discovered, installed, updated, and removed on mobile information devices.
- **End-to-end security:** MIDP provides a robust security model [15] built on open standards to protect the network, applications, and mobile information devices. The use of HTTPS leverages existing standards such as Secure Sockets Layer (SSL) and Wireless Transport Layer Security (WTLS) to allow the transmission of encrypted data.

2.7.5 Programming Architecture of MIDP 2.0

The MIDP is designed to operate on top of the Connected Limited Device Configuration (CLDC). At a high level, the MIDP [30] specification assumes that

the MID is limited in its processing power, memory, connectivity, and display size.

The MIDP 2.0 [11] specification is based on the MIDP 1.0 specification and provides backward compatibility with MIDP 1.0 so that MIDlets written for MIDP 1.0 can execute in MIDP 2.0 environments. The MIDP 2.0 specification was designed assuming only CLDC 1.0 features; it will also work on top of CLDC 1.1, and presumably any newer versions. It is anticipated that most MIDP 2.0 implementations in future will be based on CLDC 1.1

The combination of CLDC and MIDP [13] provides a complete environment for creating applications on cell phones and simple two-way pagers shown in Figure 2.5. The core of a MID Profile is a MIDlet application. The application extends the MIDlet class to allow the application management software to control the MIDlet, retrieve properties from the application descriptor, and notify and request state changes.

The application management software can manage the activities of multiple MIDlets within a runtime environment. In addition, the MIDlet can initiate some state changes by itself, and notify the application management software of those changes.

MIDP APIs for the user interface: These APIs are designed so that interaction with the user is based around a succession of screens, each of which presents a reasonable amount of data to the user. Commands are presented to the user on a per-screen basis. The APIs allow the application to determine what screen to display next, what computation to perform, and what request to make of a network service.

MIDP APIs for handling the database: These APIs organize and manipulate the devices database, which comprises information that remains persistent across multiple invocations of the MIDlet.

The underlying CLDC API is used to handle strings, objects, and integers. A subset of the Java 2 API is also provided to handle I/O and network communications.

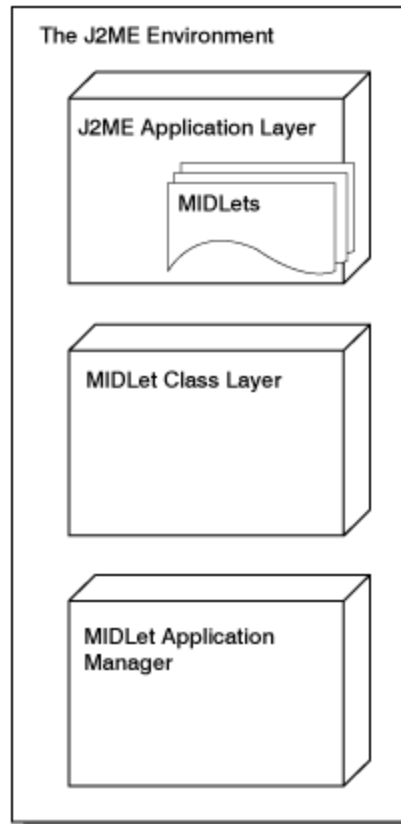


Figure 2.5: Programming Architecture of J2ME [25]

2.7.6 MIDlet

A MIDlet is a MIDP application. The application must extend the MIDlet class to allow the application management software to control the MIDlet, retrieve properties from the application descriptor, and notify and request state changes. All MIDlets extend the MIDlet class – the interface between the runtime environment (the application manager) and the MIDlet application code.

The MIDlet class provides APIs for invoking, pausing, restarting, and terminating the MIDlet application. The application management software can manage the activities of multiple MIDlets within a runtime environment. In addition, the MIDlet can initiate some state changes by itself, and notify the application management software of those changes.

2.7.7 Event handling in J2ME

Event handling in J2ME, in contrast to event handling on the desktop version of the Java platform, is based around a succession of screens. Each screen carries a certain small amount of data.

Commands are presented to the user on a per-screen basis. The Command object encapsulates the name and information related to the semantics of an action. It is primarily used for presenting a choice of actions to the user. The resulting command behavior is defined in a Command Listener associated with the screen. Each Command contains three pieces of information: a label, a type, and a priority. The label is used for the visual representation of the command; the type and priority are used by the system to determine how the Command is mapped onto a concrete user interface.

2.7.8 User Interface Design in MIDP

MIDP 2.0 provides UI classes in two packages, `javax.microedition.lcdui` and `javax.microedition.lcdui.game`, where `lcdui` stands for liquid crystal display user interface (LCD UI). Although it maintains a constrained profile, the MIDP API [30] provides a complete set of UI elements.

- **Alert:** An alert is a screen that informs the user about an exceptional condition or error. For Example, in the stock application, the alert lets the user know that a specified stock had reached a predetermined price.
- The alert screen can handle both text and images. There are two variations of alert screens: timed and modal. The timed variation allows the alert to pause for a certain period of time before proceeding to the next screen on its own. The model variation requires input from the user before it can proceed; the user must initiate a command (for example, press a button) for the screen to go away. If there is not enough space to display all of the timed alerts, and the user is forced to scroll the screen, the timed alerts can be turned into modal alerts.
- **Choice:** A Choice implements a selection from a predefined number of choices. Each element of a Choice is composed of a text string and an optional

image. If the application provides an image, the implementation may choose to ignore the image if it exceeds the device's display capacity. If the implementation displays the image, it is displayed adjacent to the text string and the pair is treated as a unit. Images within any particular Choice object should all be of the same size, because the implementation is allowed to allocate the same amount of vertical space for every element. If an element is too long to be displayed, the implementation will enable the user to see the whole element. If this is done by wrapping an element to multiple lines, the second and subsequent lines indicate to the user that they are part of the same element, not a new element. After a Choice object has been created, elements may be inserted, appended, and deleted.

- Choice Group: A ChoiceGroup is a group of selectable elements intended to be placed within a Form. The group may be created with a mode that requires either single or multiple choices to be made. The implementation is responsible for providing the graphical representation of these modes and must provide visually different graphics for various modes. For example, it might use “radio buttons” for the single-choice mode and “checkboxes” for the multiple-choice mode.
- Form: A Form is defined as a screen that contains an arbitrary mixture of items: images, read-only text fields, editable text fields, editable date fields, gauges, and choice groups. In general, any subclass of the Item class may be contained within a Form [11]. The implementation handles layout, traversal, and scrolling. None of the components contained in the Form has any internal scrolling; all contents scroll together. This differs from the behavior of other classes, the List for example, where only the interior scrolls.
- List: A List provides a list of choices. The List class is defined as a screen containing a list of choices. Most of the behavior is common with the ChoiceGroup class, and the common API is defined in the interface Choice.
- StringItem: This simple class extends Item by allowing a string to be put on a form. A StringItem is display only; the user cannot edit the contents. Both the

label and the textual content of a `StringItem` may be modified by the application.

- **TextBox:** A `TextBox` is a screen that allows the user to enter and edit text. A `TextBox` has a maximum size or capacity – the maximum number of characters that can be stored in the object at any time. This limit is enforced when the `TextBox` instance is constructed; when the user is editing text within the `TextBox`, and when the application program calls methods on the `TextBox` that modify its contents.
- **TextField:** Like the `TextBox`, a `TextField` is an item that may be placed within a `Form`, and has a maximum size determined by the maximum number of objects that can be stored in the object at any time. In a `Form`, multiple `TextFields` can be used to input data such as the stock's original purchase price and the number of shares held by the user allows the user to enter and edit text. Multiple `TextFields` can be placed in a `Form`.
- **Ticker:** This class is particularly suited for the developer's application. `Ticker` implements a ticker string, a piece of text that runs continuously across the top portion of the display. The direction and speed of scrolling are determined by the implementation.

2.6.9 CLDC API

MIDP contains the following packages, the first three of which are core CLDC packages, plus three MIDP[30] specific packages. The CLDC [27] `java.lang` package is a subset of the J2SE `java.lang` package. The most notable omissions compared to J2SE are floating point operations and, in particular, floating point (`Float`) and double precision (`Double`) classes.

The core run-time classes for the `java.lang` package are:

- `Class` -- Represents classes and interfaces in a running Java application.
- `Object` -- As in J2SE, `Object` is the base class of all Java objects.
- `Runtime` -- Provides a way for a Java application to interact with the run-time environment in which it is running.
- `System` -- Provides several static helper methods, as with J2SE.

- Thread -- Defines a thread of execution for a Java program.
- Throwable -- The superclass of all errors and exceptions in the Java language.

The core data type classes in the java.lang packages are:

- Boolean -- Wraps the boolean primitive data type.
- Byte -- Wraps the byte primitive data type.
- Character -- Wraps the char primitive data type.
- Integer -- Wraps the int primitive data type.
- Long -- Wraps the long primitive data type.
- Short -- Wraps the short primitive data type.

The helper classes for the java.lang package are:

- Math -- Contains methods for performing basic mathematical operations. Note that all the methods manipulating floating point values are omitted, leaving only the abs(), min(), and max() methods on integers and longs.
- String -- Represents String objects in Java, as in J2SE.
- StringBuffer -- Represents a string that can be modified, as in J2SE.

The CLDC API [27] contains many of the commonly used input classes from J2SE. In particular, the CLDC java.io package includes the following classes:

- ByteArrayInputStream -- Contains an internal buffer representing bytes that may be read from an input stream.
- DataInput -- An interface that provides for reading bytes from a binary input stream and translating them into primitive Java data types. DataInputStream provides an implementation of this interface.
- DataInputStream -- Allows an application to read primitive Java data types from the underlying input stream in a platform-independent way.
- InputStream -- An abstract class that is the superclass of all classes representing an input stream of bytes.

- `InputStreamReader` -- Reads bytes and translates them into characters according to a specified character encoding.
- `Reader` -- An abstract class for reading character streams.

The CLDC API [27] contains many of the commonly used output classes from J2SE. In particular, the CLDC `java.io` package includes the following output classes:

- `ByteArrayOutputStream` -- Implements an output stream where data is written into a bytes array.
- `DataOutput` -- An interface that provides for writing primitive Java data types to a binary output stream. `DataOutputStream` provides an implementation of this interface.
- `DataOutputStream` -- An output stream that allows an application to write primitive Java data types in a portable way.
- `OutputStream` -- An abstract class that is the superclass of all classes representing an output stream of bytes.
- `OutputStreamReader` -- Given characters, translates them into bytes according to a specified character encoding.
- `PrintStream` -- Adds a convenient way to print a textual representation of data values.
- `Writer` -- An abstract class for writing character streams.

The `java.util` collections classes supported by CLDC are:

`Enumeration` -- An interface that allows the calling routine to iterate through a collection of items, one at a time.

- `Hashtable` -- Implements a hashtable, which maps keys to values.
- `Stack` -- Represents a last-in-first-out (LIFO) collection or stack of objects.
- `Vector` -- Represents a resizable "array," or vector, of objects.

The remaining `java.util` classes supported by CLDC include date and time classes and the `Random` utility class. These are summarized in the following table.

- Calendar -- An abstract class for getting and setting dates using a set of integer fields such as YEAR, MONTH, DAY, and so on.
- Date -- Represents a specific date and time, with millisecond precision.
- Random -- Utility class used to generate a stream of random int or long values.
- TimeZone -- Represents a time zone offset, and also handles daylight savings adjustments.

So far, all the classes we have seen in the CLDC API have been a subset of the J2SE API. CLDC also includes one additional package -- the `javax.microedition.io` package. The only class defined in this package is the Connector class, a factory class that contains methods to create Connection objects, or input and output streams. Connection objects are created when a class name is identified dynamically. A class name is identified based on the platform name, as well as the protocol of the requested connection.

In addition to the generic connection factory class, the `javax.microedition.io` package also contains the following connection-oriented interfaces:

- Connection -- Defines the most basic type of connection. This interface is also the base class for all other connection interfaces in this package.
- ContentConnection -- Defines a stream connection over which content is passed. Datagram -- Defines a generic datagram interface.
- DatagramConnection -- Defines a generic datagram connection and the capabilities it must support.
- InputConnection -- Defines a generic input stream connection and the capabilities that it must support.
- OutputConnection -- Defines a generic output stream connection and the capabilities that it must support.
- StreamConnection -- Defines a generic stream connection and the capabilities that it must support.

- `StreamConnectionNotifier` -- Defines the capabilities that a stream connection notifier must have.

MIDP applications are called MIDlets, as are Java class files of a specific structure that are packaged with a descriptor file and an archive file (JAR) to comprise a MIDlet suite.

2.7.10 MIDP API

MIDP [30] encompasses the four core CLDC packages (`java.lang`, `java.io`, `java.util`, and `javax.microedition.io`), plus the following three MIDP-specific packages:

- `javax.microedition.lcdui` -- Defines classes that provide for control over the UI. This package includes both the high-level UI classes (such as `Form`, `Command`, `DateField`, `TextField` and more), as well as the low-level UI classes (allowing low-level control over the UI)
- `javax.microedition.midlet` Contains one of the main MIDP classes, the `MIDlet` class, which provides MIDP applications access to information about the environment in which they are running.
- `javax.microedition.rms` Defines a set of classes that provide a mechanism for MIDlets to persistently store, and later retrieve, data.

MIDP also adds four new classes, shown below, to the core CLDC packages.

- `java.util.Timer` -- Used to schedule tasks for future execution in a background thread.
- `java.util.TimerTask` -- Used by the `java.util.Timer` class to define tasks for later execution in a background thread.
- `javax.microedition.io.HttpConnection` -- An interface that defines the necessary methods and constants for an HTTP connection.
- `java.lang.IllegalStateException` -- A `RuntimeException` that indicates that a method has been invoked at an illegal or inappropriate time.

2.8 How Java Applications Run on Mobile Phones

Applications developed for the J2ME CLDC Java platform are called MIDlets. They are downloaded to the device in the form of two files:

1. The Java Archive (JAR): The JAR is an archive that contains the following files: JAR manifest, class files, and supporting files. The JAR manifest is a text file that contains various attributes like the MIDlet name and vendor. Class files are the preverified class files of the MIDlet (pre verification will be discussed later). Supporting files are any files needed by the application like graphic files for instance. A JAR file can contain more than one application (MIDlet). This group of MIDlets is called a MIDlet suite.
2. The Java Application Descriptor (JAD). The JAD on the other hand, is a text file with several attributes like MIDlet name and the MIDP version needed to run the MIDlet.

2.9 J2ME Web Services API (WSA)

Web Services are loosely coupled reusable software components that semantically encapsulate discrete functionality and are distributed and programmatically accessible over standard Internet protocols.

The technological evolution of Post PC era allows using mobile devices for accessing Web Services [7]. Therefore, the mobile device manufacturers, the service platforms providers and of course the developers have a great opportunity for building high quality applications and services for the massive market that uses Web Services. However, not all Web Services may be available for accessing from mobile devices, due to high processing and memory requirements they may demand. As Web Services are XML (eXtensible Markup Language) based, SOAP (Simple Object Access Protocol) messages used to be verbose, which implies burdensome parsing tasks. On the other hand, not all Web Services defined data types are supported for mobile devices. Therefore, it is required to define access architectures in order to mobile devices may overcome these constraints. Developed within the Java Community Process as Java Specification Request

(JSR) 172, WSA [28] extends the J2ME to support Web Services. This web service [7] architecture has three elements shown in Figure 2.6

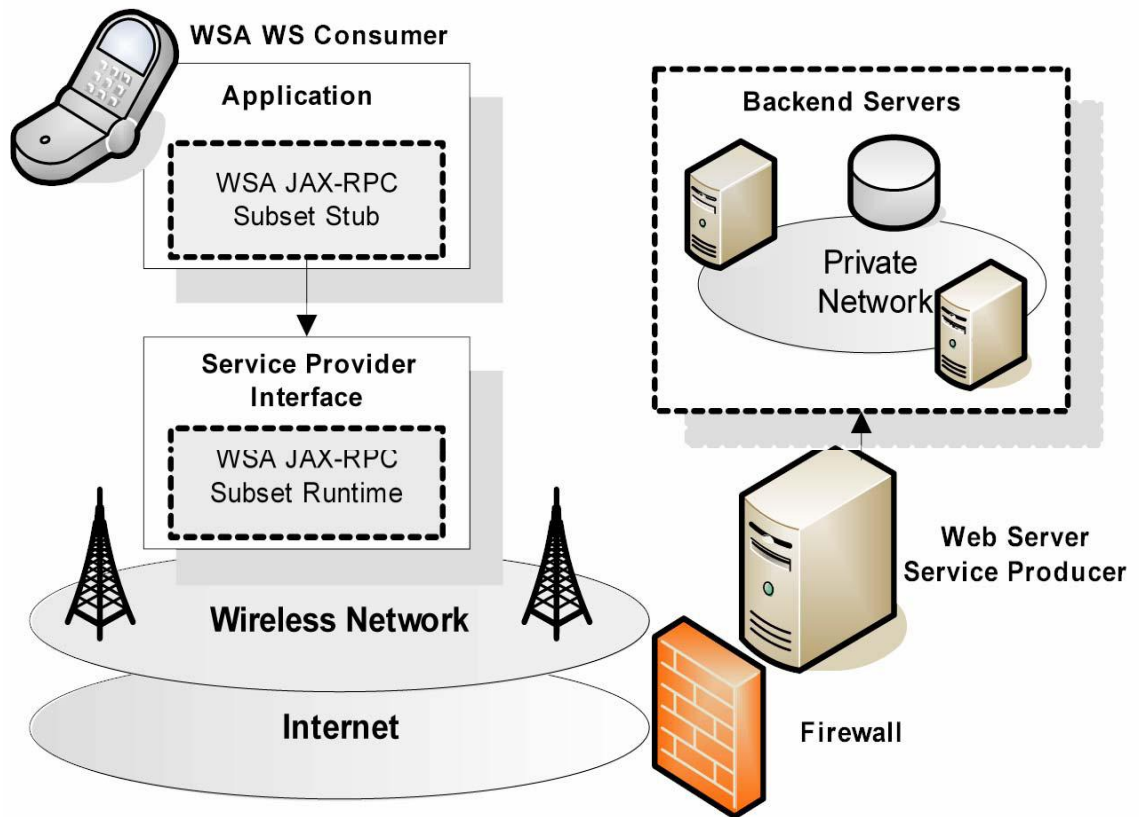


Figure 2.6: WSA working [19]

2.10 Language Transliteration

The phenomenon of mapping word by word of source language to target language is known as Transliteration. It is not truly a language transcription but approaches towards it. The source language word is supplied as input while the output is generated as the target language word. The source language in studied systems is roman English, while the target language is Gurmukhi Punjabi language.

The most spoken and understandable language of the north India and NRI Diaspora is Gurmukhi Punjabi language while they do not know exactly how to write the words or sentence in Gurmukhi Punjabi language. Till now the best

option for such users was to use the Punjabi phonetic keyboard in which most of the Roman keys were mapped to phonetically nearest sounding Gurmukhi characters. But these keyboards had their shortcomings, as not all the keys could be mapped.

The language is also not standardized for the informal use. It varies from every village to village. This adds to problems for language transliteration software development. Another problem is that all the sounds of Punjabi are not properly represented in English. A user can use same English character to represent more than one Gurmukhi character. Thus 't' could represent both t and q in the Gurmukhi script. The challenge for the software is to automatically map the Roman character to an appropriate Gurmukhi character.

To resolve the problem the transliteration is one of the viable solutions. The most popular software in this area is 'AKHAR' [21] Punjabi word processor. The software uses artificial intelligence techniques to cleverly guess the most appropriate word/alphabet and presents the other phonetically similar words sorted on their relevance to the user in a suggestion box. The user can choose the appropriate suggestion if the software cannot guess the word correctly. Though it has been observed that in more than 95 per cent of cases, the software can correctly guess the word.

The artificial intelligence based fuzzy logic techniques used by the software are backed up by a rich collection of lexical resources, including a Punjabi corpus of more than 74 lakh Punjabi words, a Punjabi Dictionary of 1.5 lakh words and an English-Gurmukhi Transliteration dictionary of more than 20,000 commonly used English words and names

Various problems have been solved by the predictive transliteration system. One can also mix English words, which are now part of the Punjabi language, such as school, college, city, thanks, lovely, etc., along with the Punjabi words. The English words can be typed according to spellings and they will be transliterated to Gurmukhi. If a word with typical spellings is not transliterated properly, it can be added to user dictionary.

2.10.1 English to Punjabi Dictionary

The description of the word in one language to another language is done in dictionary that is used to elaborate the meaning and the appropriate usage of the word.

The abundance of online dictionary makes the complex task of search and find of meaning of words in just few clicks. The chiefly found and studied dictionary are given below

1. Electronic Bilingual English-Punjabi dictionaries, Punjabi-English and English-Punjabi dictionaries have been incorporated in Akhar, which can be invoked by clicking on any Punjabi or English word and the meaning of the word is displayed in a new window. Optionally the user can also browse through the dictionaries or type a word in the edit box to get its meanings. The dictionaries can prove very helpful for persons working on Punjabi-English or Punjabi-English translation. The user can just select one of the meanings and replace it with original word and no typing is needed. For example, if a translator is translating a typed English text into Punjabi and if the word Language has to translated, then the user can double click on the word and instantly the English-Punjabi dictionary will be opened and the Punjabi meanings of the word are displayed. The user can select the appropriate Punjabi word and click the Replace button and the word Language will be replaced by BaSa in the text. Similarly the Punjabi-English dictionary can prove to be very useful for Punjabi to English translation.
2. Punjabi English online dictionary Khoj[22] by Punjabi University Patiala.
3. The link www.ijunoon.com that provide online dictionary translation.
4. Punjab online that translate English to Punjabi word.

2.10.2 Techniques of Language Translation

The action of changing one language to another is defined as the translation. The use of machine to accomplish this task is defined as machine language translations.

Different types of the methods used in this phenomenon are

- Rule Base Translation: It includes Transfer-based machine translation, Interlingual machine translation, And Dictionary-based Machine Translation. The Transfer based and Interlingual method use the intermediately representations. This intermediately representation capture the meaning of the original text in order to generate correct translation. This Dictionary based translation as the name suggests uses the dictionary conversion of the sentence.
- Statistical Translation: Some statistical method is used on bilingual corpus to generate the required results.
- Example based Translation: The Knowledge base of bilingual corpus is used at the run time but the translations are done by analogies. It can be viewed as cased base reasoning approach of machine learning.

The above said services already are in use on the computer system where the resources are not the problem. The accessibility to the computer is also not a problem but there is an issue of instant availability. The convergence of the existing facility with the emerging technology is the goal of this thesis. The bridge technology is studied and that can be used to attach the existing computer transliterating or translating software with mobile phones.

We have studied the literature in this chapter and on base of that we have formulated the problem in next chapter. We have created an application that operates on the mobile phones using J2ME environment and Direct Translation Methodology.

Chapter 3

Problem Statement

The problem of language pertains in every region of the world. The globalization has come up with the need for different languages in computer system. The Punjabi people travel most in the world and they face the problem of understanding the universal language English and NRI Punjabi do not understand the Punjabi language.

Punjab has biggest NRI population and they frequently visit to their motherland. The new generation of these NRI people does not have understanding to the Punjabi language. When a NRI Punjabi sits at roadside Dhaba and asks for tea or food he/she would send a SMS to the translation service [2] and receive the SMS as output in the desired language (Punjabi). Other application is that we implemented an interface in mobile that is used to transliterate the English text to Punjabi text.

3.1 Gap Analysis

The localization was done on the computer systems by developing language translation system like SYSTRAN [12], which translates European languages. For using such application, user has to write sentence in source language and specify target language, the application sends this request to server and bring the equivalent sentence in target language. These softwares are effectively in use by many companies, organization and individual for various purposes.

This type of software's may certainly be proved to be more useful if become available on mobile phone for some users like tourists, salesman, students etc.

Some of the gaps in the existing technologies are as follows:

1. There is no system which can perform translation of text from Punjabi language to other languages. Here we concentrated on the translation of Gurmukhi Punjabi language from Roman English. The existing system like 'Akhar'[21] is computer based transliteration software. The other online systems English to Punjabi dictionary are created by Punjab online,

ijunoon and online dictionary by Punjabi university [22]. All these systems are computer based facilities that can be used on computer systems only.

2. At present there are no existing applications that can perform translation/transliteration of English language to Punjabi language through mobile phones.
3. The existing system is used in computer and the mobility of present system is limited.
4. The cost for using this technology is high. The mobility can be gained by adding cost and weight to the system that is using Laptops.
5. The existing system is concentrating on internet technology and is not WAP [4] enable.

3.2 Methodology

We proposed the implementation of the localization that is a regional language at mobile phones by use of MIDlet, Web services. The Gurmukhi Punjabi is not available on mobile systems. Our effort is to provide the local language by implementing the translation/transliteration at mobile hand set. This can be done with the help of java programming language on mobile phone. It is used to develop the application of translation/transliteration at mobile set. Web Service is another mean of connecting mobile with the existing computer based translation services. The SMS [26] from mobile can be sent to translation service as input and in return get the translated text back.

The existing machine language translation is focused on computer systems. These software's are resource consuming. We target the mobile phones to connect the existing systems and providing desired language at any desired location to the user.

3.3 How it Help?

Current research work is focused on computer based translation/transliteration. This study reveals the work that is already there can be incorporated in mobile

phones also. The solution to problem will help the NRI Punjabi to understand and communicate in Punjabi language. It will also help the people of other state and countries to understand the text written in Punjabi with facility of dictionary and translation.

Design and Implementation

The java is used as development environment. The application that is developed in J2ME/BREW can run over both the smart and featured phone. The J2ME is attached feature in the Symbian operating system which controls 60% market of the mobile phones. The java on each type of the mobile phone makes the J2ME a very valuable tool for development.

The Sun Java Wireless Toolkit for Connected Limited Device Configuration (CLDC) (formerly the J2ME Wireless Toolkit) is a state-of-the-art, award-winning toolkit for developing wireless applications using the Java programming language. The Java Wireless Toolkit for CLDC functions as a standalone development environment, or it can be used with an integrated development environment (IDE), such as the NetBeans.

Mobility Pack for CLDC the SUN java provides toolkit [6] called sun_java_me_sdk-3_0-ea-win. This toolkit provide environment for CLDC/MIDP application development on mobile.

Netbeans 6.0 [17] is a free, open-source Integrated Development Environment for software developers. This development environment helps the user to create professional desktop, enterprise, web, and mobile applications with the Java language, C/C++, and even dynamic languages such as PHP, JavaScript, Groovy, and Ruby.

4.1 Features of Netbeans

The Netbeans [17] is a collection of rich development, editing, design features for mobile systems. It has been dual licensed under the Common Development and Distribution License (CDDL) and the GNU General Public License version 2. The Integrated Development Environment is rich in features as follows:

- Phones, Hand-Held Devices, and Set-Top Boxes: Create, test and debug applications for the Mobile Information Device Profile (MIDP) 1.0, 2.0, 2.1

(MSA), the Connected Limited Device Configuration (CLDC) 1.0 and 1.1, and the Connected Device Configuration (CDC).

- Visual Mobile Designer (VMD): Develop GUIs rapidly with the Visual Mobile Designer (VMD): Drag and drop components like wait screens, login screens, file browsers, an SMS composer, and splash screens are included. The Analyzer tool helps you decrease file size by identifying unused components for removal, and it also checks for MIDP 1.0 compliance. The VMD also makes GUI localization easier.
- SVG Component Palette: Select basic shapes from the SVG image palette to get started and use the other SVG components in the palette such as SVGForm, Button, CheckBox, TextField, Slider, and more to create rich content interfaces for your mobile applications.
- Mobile Game Builder: Develop mobile games with a visual editor designed for the MIDP 2.0 Game API. The API allows you to create game scenes with sprites on a game canvas using tiled layers and layer management.
- Access to Web Services: Write applications that access web services directly from JSR-172-enabled phones. Use the Wireless Connection Bridge to access web services or other server-side data via servlets from any device using MIDlets.
- Modular Build System: Enhance the Mobility pack by plugging-in your own components. You can add new platform types, deployment types, configurations providers, and more.
- Integrated Tools: The IDE integrates tools for JUnit 1.1.0 testing, MIDlet signing, certificate management, automatic code obfuscation with ProGuard 4.2, integrated over-the-air (OTA) emulation, push registry emulation, WMA emulation for SMS and CBS messages, new deployment methods, and Wireless Messaging and Multimedia APIs. You have complete control over these tool using standard Apache Ant scripts.
- New Integrated UI for CLDC/MIDP and CDC development. The Mobility pack now supports the project properties previously available only for CLDC/MIDP projects. These include project configuration support for device

fragmentation, integrated obfuscation and optimization support, and multiple deployment options, all built on Apache Ant for easier coding and management.

4.1.1 The Wireless Toolkit Features

The Sun Java Wireless Toolkit is a collection of tools and device emulators that support the development of Java applications that run on devices compliant with the Mobile Device Information Profile (MIDP) [18]. The Sun Java Wireless Toolkit is installed in the NetBeans IDE as the default emulator. The Java Wireless Toolkit emulator, based on Java ME CLDC [27] and MIDP reference implementations, is fully compliant with the Technology Compatibility Kit (TCK) for the supported Java Specification Requests (JSRs)

- The Wireless Toolkit can simulate several devices that take on a variety of form factors cell phones, pagers, and palmtops. This helps us to learn what type of experience users can expect and verify the portability of our application across different devices.
- The Wireless Toolkit supports tracing and debugging with run-time logging of various events, such as garbage collection, class loading, method calls, and exceptions. We can also perform source-level debugging with an IDE while an application runs in the Wireless Toolkit.
- The Wireless Toolkit enables us to collect information that we can examine to optimize the performance of your MIDP applications. Performance tuning utilities include profiling methods and monitoring memory usage and network traffic. Also included is the ability to adjust the speed settings for graphic rendering and refreshing as well as VM speed emulation and network throughput.
- The new Java Wireless Toolkit includes support for the Mobile Service Architecture platform as well as Nokia's Scalable Network Application Package (SNAP) Mobile API.
- Creating a MIDP Application Using the Visual Mobile Designer. The NetBeans IDE provides a wizard that enables us to quickly create a MIDP

project. When creating the project, we can choose to develop our application in the Visual Mobile Designer (VMD) or in the Source Code Editor. Using the Visual Mobile Designer gives us the ability to graphically plan out the flow of the application and design the screens the application uses. The Visual Mobile Designer (VMD) is a graphical interface within NetBeans Mobility that enables us to design mobile applications using drag and drop components. The VMD allows us to define the application flow and design our GUI using the components supplied by the IDE or components we design ourselves.

4.1.2 Installation and Configuring of Netbeans 6.0 with J2ME

Required Software: Before installing the IDE, the Java SE Development Kit (JDK) 5 Update 14 (version 1.5.0_14) or newer (including JDK 6 Update 4) must be installed on system. We had latest version of Java SE Development Kit (version 1.6.0_13). The Installer shows the following options:

- Web & Java EE. Provides tools for developing Java SE, Java EE, and web applications. This download option also includes the GlassFish V2 UR1 application server and Apache Tomcat software.
- Mobility Supports Java ME application development and includes the Mobility pack for creating MIDP applications for mobile devices. Some Mobility features, such as the Wireless Connection wizard, require features that are available in the Web & Java EE pack.
- Java SE. Provides all standard features for Java SE development. Support for NetBeans Plugin Development is also included.
- Ruby. Provides tools for Ruby development and supports Rails and JRuby.
- C/C++ Supports development in the C/C++ language.
- Automatically the installer checks the version of java available at system and if it found it goes to next option page else it will give alert to us that java is not installed, install it first.
- The option page will ask the user to select the packages he/she want to install like Java EE, Java Mobile or Java SE.

- We selected the Java Mobile install option along with the default selected packages (Java EE, IDE Environment).
- Then, we started the Netbean 6.0 application.

4.1.3 Creating the MIDP/CLDC Application

The java mobility pack in Netbean can be designed and implemented quickly. Creating an application a mobile application in Netbean 6.0 involves very swift and small steps that are listed below.

- Choose File > New Project (Ctrl+Shift+N). Under Categories, select Mobility. Under Projects, select MIDP Application and click Next.
- Edit the Project name in the Project Name field. Use the default Project Location, or change it to the directory you prefer on your system.
- Check the Set as Main Project and Create Hello MIDlet check boxes (both are checked by default). Click Next.
- Select the Sun Java Wireless Toolkit 2.52 for CLDC as the Emulator Platform and use the remaining defaults. Click Next.
- The last click is Finish and this will create the project file and its configuration files in the project folder.

4.1.4 Life Cycle of MidLet

The application developed in this thesis is based on Mobile information device profile and these are called MIDlet. The MIDlet of Java 2 Micro Edition is relevant to the Applet of Java 2 Standard edition. Every MIDlet must have three methods that contitute the life cycle of MIDlet:

- startApp: startApp method is called when the MIDlet is started. After startApp, the MIDlet is in Active state and the AM (Application management Software) allows it to hold resources. If a runtime exception occurs during startApp, the MIDlet will be destroyed immediately.
- pauseApp: Usage of pauseApp is not supported in S60 and Series 40 platforms. It is still required by the MIDP specification and must therefore be included in the MIDlet although it can not be called at any point.

- `destroyApp`: `destroyApp` signals the MIDlet to terminate and places it in *Destroyed* state. During termination, all the MIDlet's resources are released and objects deleted. The MIDlet has five seconds to handle the `destroyApp` call, after which the AMS closes the application itself.

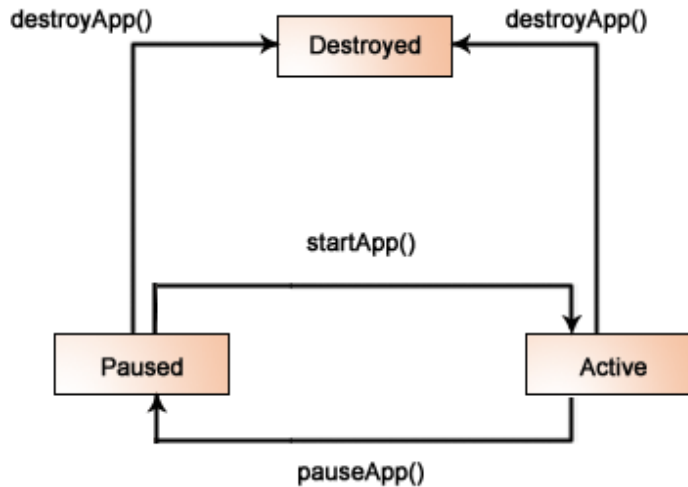


Figure 4.1: MIDlet Lifecycle [30]

By default MIDlet is in the paused states. When the application is executed by default `startApp` method will call and when close the application the `destroyApp` method will be called. But when the constructor is not null type then it will be executed firstly.

4.2 Application Development Process

We used the Visual Mobile Designer of Netbeans. The VMD develops GUIs components rapidly. With the Visual Mobile Designer (VMD) Drag and drop components like wait screens, login screens, file browsers, an SMS composer, and splash screens are included. The Analyzer tool helps us to decrease file size by identifying unused components for removal, and it also checks for MIDP 1.0 compliance. The VMD also makes GUI localization easier.

This application is simple dictionary based translating solution for mobile. The input is given by the user as the source language that is roman English and the output is target language as Punjabi. The translation is direct translation. A few daily used sentences are stored in the file and direct searching for the sentences is

done. The Flow diagram in figure 4.2 shows the various Components used in the Designing of the application.

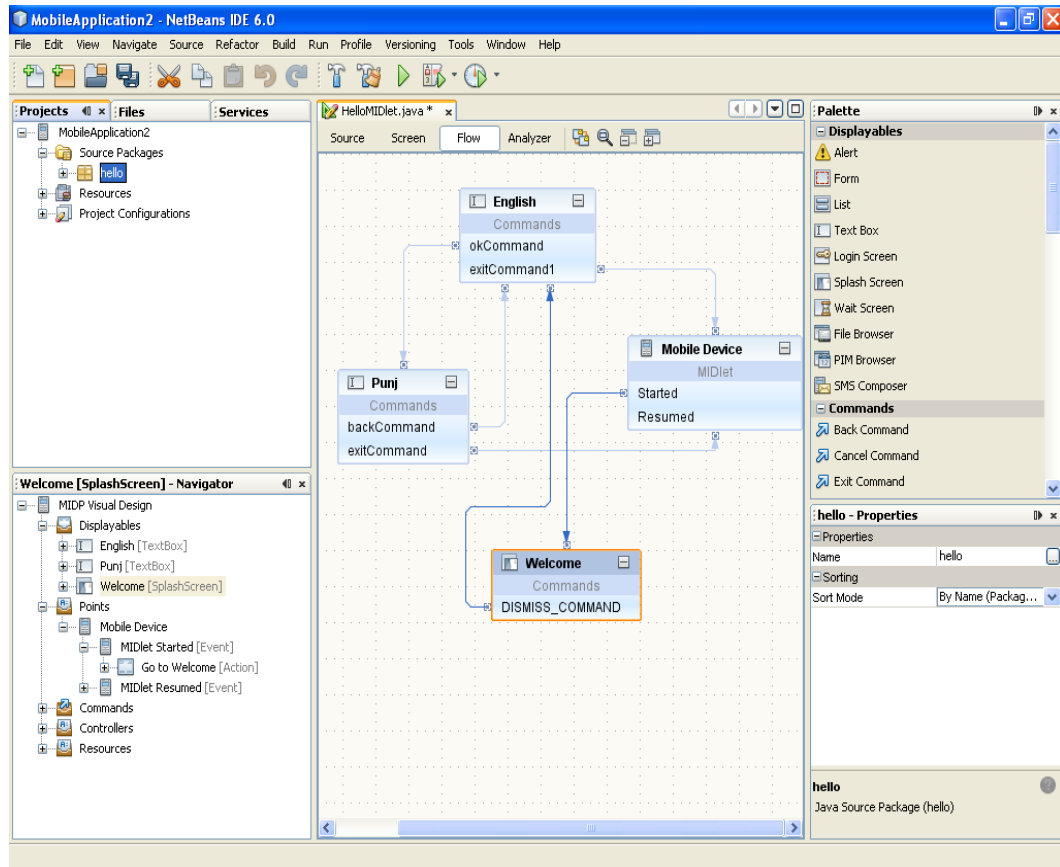


Figure 4.2: Netbean Mobility Application

4.2.1 Work Flow Diagram of the application

Visual mobile designer is used to Design the application. The VMD of Netbeans is collection of rich User interface tools which is used in designing the application. The snapshot 4.3 shows the Visual Mobile designer and the application flow.

- The new application begins with the MIDlet device screen and it has states started and resumed. By default the mobility application begins with this Mobile device on VMD Flow Diagram
- The other components that are dragged from palette of displayable are a) Splash Screen b) Text box1 c) Text box2

- The commands used for event handling added to these displayable are Dismiss Command with Splash Screen, Ok and Exit command with Text Box1, Back and Exit command with Text Box 2.
- After adding command the flow links from one text box to another and with other components are formed as shown in the Figure 4.3.
- The beauty of Netbeans is that with completion of Design with help of VMD it itself create the code for the application and we only need to add the constraints, connection with the file, the algorithm for search of words in the file.

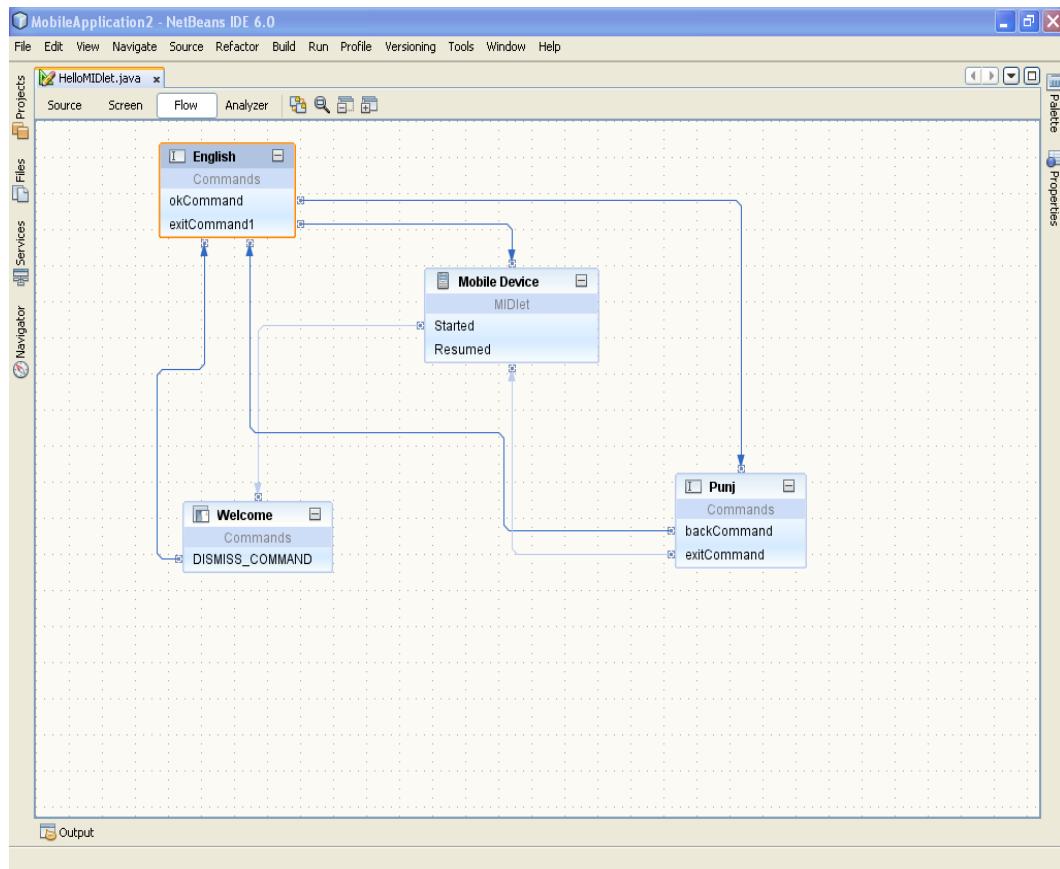


Figure 4.3: Workflow Diagram of application

The UI (User Interface) components and their working in the application are

- Mobile Device: The Emulator used in the application. In this application we used QWERTY emulator from sun. The Mobile block in diagram shows the State of MIDlet *i.e* Started or Resumed.

- Splash Screen: When the application is launched the splash screen display the starting point and tell the user about the application.
- Textbox1 (Labeled as English): The text by the user as source language is inputted as roman English. Controls with this Screen are OK Command and Exit Command. The Ok Command transfers Screen to Next Screen. The Exit Command exist the application.
- Textbox 2 (Labeled as Punj): The system then search for the text in the file and if it found the text it give the output as target language that is Punjabi in the text box else it displays not found text. The Back command transfers the control back to the English *i.e* TextBox1. The Exit Command exist the application.
- The file connection is made to the file exists at Computer systems using input output package of Java microedition.
- For the search of the string from the File contain translation of the English sentence to Punjabi sentences that is attached with the application. We have used simple linear search algorithm to implement search of the string from the file.

Chapter 5

Results

The pervious chapter disclosed about the tool used in this thesis to design and implement the application. In this chapter we look into the results given by the application for the different type of the cases. When we run the application through Netbeans, the emulator starts and asks the user to launch the application. The application begins with splash screen on emulator shown in Figure 5.1 that shows the user what it actually does.

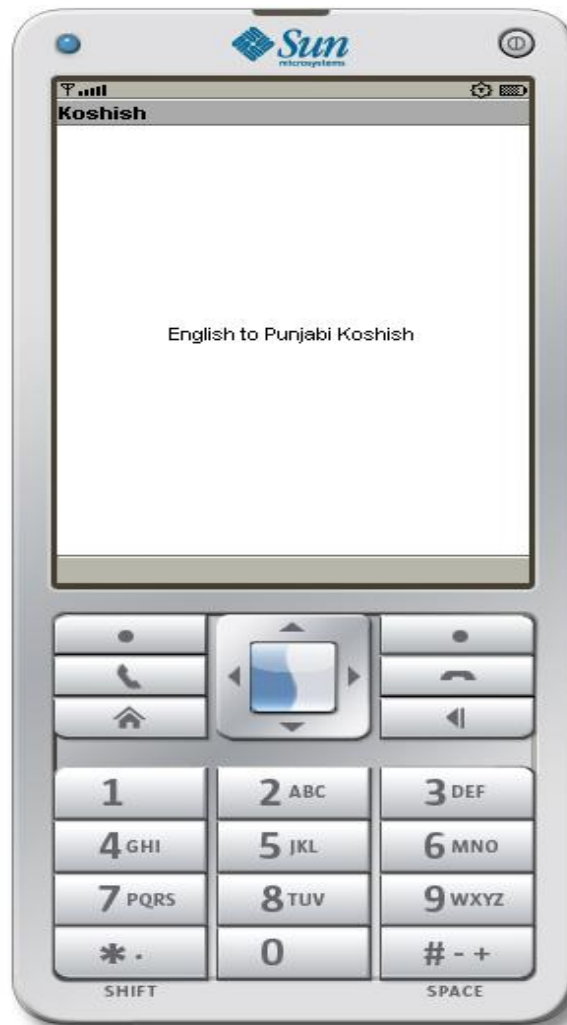


Figure 5.1 : Starting Screen of Application

5.1 Case 1

In this case we assumed that the user enter the single word of English and wants the output or dictionary translation of that word in Punjabi. The user enter the word in English shown in figure 5.2 “water” and after entering it when the user press OK command, the control transfers to next screen with text box named Punj along with the output as dictionary meaning of word in Punjabi *i.e* PAANI as shown in Figure 5.2.



Figure 5.2: Application used as Punjabi Dictionary

5.2 Case 2

When the user has entered the text that is more than one word, the application will search for the complete translation of the sentence from the file. The file also consists of commonly used Punjabi phrases and that can be used directly by the application for translation. Now when the user press ok the complete translated text would appear in the emulator screen as shown in the figure 5.3.

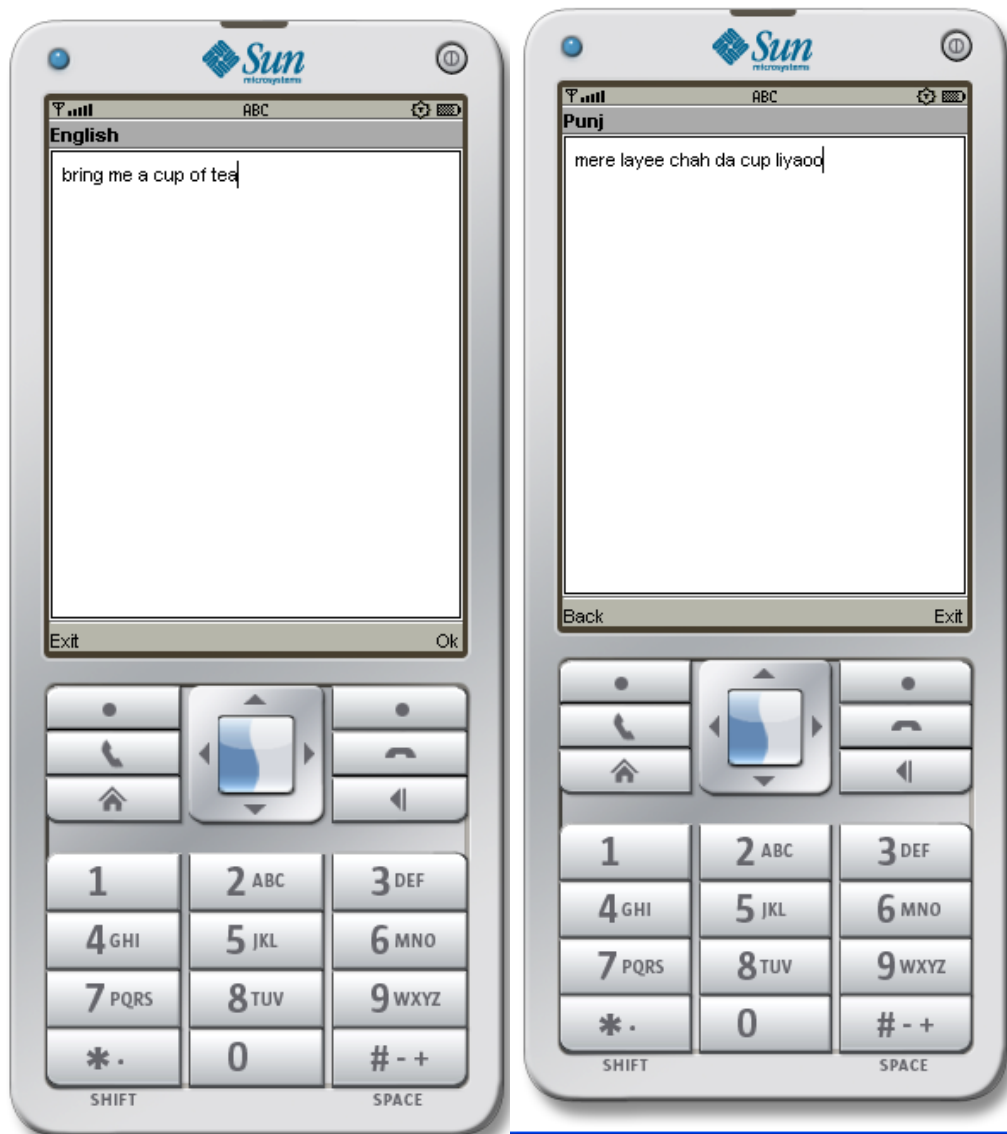


Figure 5.3: Snapshot of Application used as Punjabi Translator

5.3 Case 3

This is the case when the user enters the text and the application does not find the text from the file as show in the Figure 5.4.The application will show the message to try again. This is the case when the string entered by user in the English text box does not have the corresponding translated string in Punjabi in our file. Language is abysmal resource of words and we have captured only a few Thousands words/ sentences in our file.

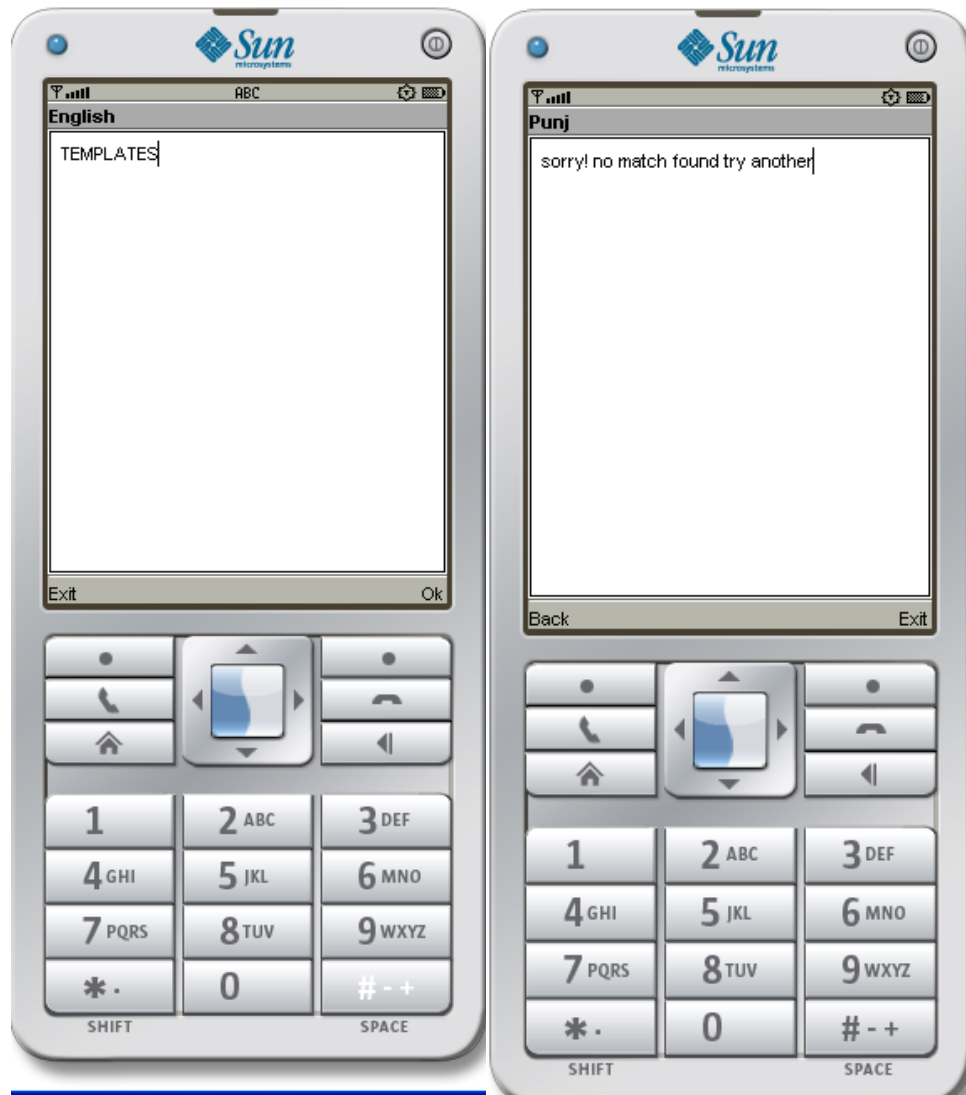


Figure 5.4: Word not Found Condition

The experimental results of the application used on mobile emulator are shown in this chapter. The next chapter concludes and gives the further scope of this thesis.

Conclusions and Future Scope

The study incite us to bridge the existing transliterating, dictionary system with the new vibrant, happening and emerging technology of Mobile phone. With the advancement of the infrastructure in wireless network and improvement in the mobile phone hardware capacity, this problem can be solved with the optimum solutions.

What has always driven the market for portable electronic devices is the unquenchable consumer demand for access to virtually unlimited amounts of information, anywhere, anytime. Wireless phones are no longer just used for voice communications. The growth of mobile devices brings new challenges and opportunity for computational support for cross language communication.

6.1 Conclusion

The mobile phones grow with advancement of the radio communication and wireless technologies. The mobile phone currently not only used for voice communication but for data transfer and data processing also. The mobile phones are the need of hour. The J2ME can be used to develop portable application on mobile phones. The J2ME environment provides the mobile interface for domain specific machine translation using MIDlet and Web Services. The study states

- Increase in the availability of translation/transliteration services if they are provided through mobile phones along with existing Computer System.
- The J2ME is tool that is rich in features like reliability, ease to use, secure and over and above portability. All mobile operating systems support J2ME.
- The other Development environment like BREW, C/C++ are very close to machine level and are specific to machine architecture.
- The use of existing translating/transliterating system with the help of Web Services by mobile phones is required. It will reuse and integrate the existing work with emerging technologies.

- Punjabi character on mobile is dependent on mobile operating system of mobile architecture. Getting Punjabi script on mobile is cumbersome and domain specific task.

This experiment is just the cog in a big wheel. It is a small effort that can be used in a language specific domain. This experiment of translation service is just a beginning in the direction of efficient, convenient, any time, anywhere machine translation. It is an endeavor to bridge the ethnical, cultural and geographical divisions between the Punjabi speaking communities. By implementing this system, new horizons for thought, idea and belief will be shared.

6.2 Future Scope

The application is on a mobile phone and the user has to download it from the source i.e. from computer System or from website. This limits the scope of utilization of the application. The translation of the words is also limited due to scarcity of space in mobile phone. The following advancements can be incorporated in the current solution:

1. We have developed the application by using dictionary based approach. The accuracy and quality of work on translation/ transliteration can be extended by implying the example based or statistical based methods of machine language translation techniques. This will improve the current searching and scope of the existing application
2. The work can be extended to translation of the English language to Punjabi language and then bridging it with mobile interface using Web Services. The Web Services shall enable the mobile phone to get the translated source language without installing an application. The short message can be used for sending the request and receiving the requested target language. The web service for the existing Punjabi translation/transliteration websites like AKHAR, KHOJ *etc* can be created and then mobile phones can connect to them.
3. The Punjabi language translation/transliteration can be extended to other languages like Hindi, Marathi, Gujrati, French *etc*.

4. The speech recognition can more effectively do on the mobile communication to overcoming the language barriers. The speech recognition technology combined with mobile computing could be used to communicate in more natural way across language barrier.

References

- [1] Andrea “Third Generation 3G Wireless” white paper
http://members.tripod.com/~srohit/wp_3g.pdf.
- [2] Avinash J. Agrawal, Manoj B. Chandak “Mobile Interface for Domain Specific Machine Translation Using Short Messaging Service” International Conference on Information Technology (ITNG'07)
- [3] By Sing Li, Jonathan Knudsen “Beginning J2ME: from novice to professional” Edition: 3, Published by Apress, 2005 pages 12-18.
- [4] Carl H. Marcussen, “Mobile Phones, WAP and the Internet”, Researcher, PhD, Research centre of Bornholm, Denmark.
- [5] Cheng-Han Tu and Bing-Yu Chen “The Architecture of a J2ME-based OpenGL ES 3D Library” Ninth International Conference on Computer Aided Design and Computer Graphics (CAD/CG 2005).
- [6] Dawn Phillips “Sun Java Wireless Toolkit for CLDC” published by Sun Micro system Available at : java.sun.com/j2me/docs/sun_java_wireless_tlkit.pdf
- [7] Eric Newcomer “Understanding Web Services: XML, WSDL, SOAP, and UDDI” published by Addison Wesley page 5-36
- [8] G. Lawton, “Moving Java into Mobile Phones,” Computer, vol. 35, no. 6, June 2002, pp. 17–20.
- [9] Gary J. Mullet “Introduction to Wireless Telecommunications Systems and Networks” Published by Delmar Cengage Learning, page 35-40.
- [10] Ian Poole “What Exactly Inside Mobile” IET Communications Engineer April/May 2006.
- [11] Java Community Process “Mobile Information Device Profile JSR 118” version 2.0 for Java 2 Micro Edition from <http://jcp.org/aboutJava/communityprocess/final/jsr118/index.html>.
- [12] Jean Senellart, Peter Dienes and Tamas Varadi, “New Generation Systran Translation System”, MT Summit VIII, Santiga de Compostela, Spain, 18-22 Sept. 2001.

- [13] Jonathan Knudsen "Wireless Java: Developing with J2ME, Second Edition" page 8-18.
- [14] Mona Baker, Kirsten Malmkjaer Contributor Mona Baker, Kirsten Malmkjaer "Routledge Encyclopedia of Translation Studies" Published by Routledge, 2001 pages 134-137.
- [15] Mourad Debbabi, Mohamed Saleh, Chamseddine Talhi and Sami Zhioua "Security Analysis of Mobile Java" International Workshop on Database and Expert Systems Applications 2005.
- [16] Natthan Eagle "Mobile Application Prototyping with Python" Entrepreneurial Programming and Research on Mobile, A 3-Day Crash Course at MIT Design Laboratory Massachusetts Institute of Technology August 2006.
- [17] Netbean 6.0 IDE Features available at <http://www.netbeans.org/features/javame/index.html>
- [18] Nicolas Lorain, Nachi Periakaruppan, Larry McDonough "Mobile Information Device Profile" white paper from <http://java.sun.com/products/midp/midp-ds.pdf>
- [19] Oscar Mauricio Caicedo Rendón, Francisco Orlando Martínez Pabón, Marlon Julián Gómez Vargas, Javier Alexander Hurtado Guaca1, Grupo de Ingeniería Telemática, Universidad del Cauca, Popayán "Architectures for Web Services Access from Mobile Devices".
- [20] Panos Asproulis "Introduction to Carbide C++" white paper published by Symbian Developer Network 2007.
- [21] Punjabi word processor available at: <http://www.akhar.org/>
- [22] Punjabi dictionary, word processor available at: <http://www.punjabikhoj.com/khojp.aspx>
- [23] Qualcomm Incorporated, "Starting with BREW" 80-D4725-1 REV.A, 27 April, 2004.
- [24] Sam Mason and Elise Korolev "Native Java ME Development on Symbian OS" Published by Symbian Developer Network. Version: 1.0 – March 2008.
- [25] Soma gosh "Architecture of J2ME" Available: <http://www.ibm.com/developerworks/library/wi-j2me/>
- [26] "SMS Tutorial" Available: <http://www.developershome.com/sms/>

- [27] Sun Microsystems “Connected Limited Device Configuration Specification Guide” version 1.1 for Java 2 Micro Edition.
- [28] Sun Microsystems. J2ME WSA Specification. Available:
<http://java.sun.com/products/wsa>
- [29] Sun Microsystems J2ME Specification.
Available:<http://java.sun.com/j2me/index.jsp>
- [30] Sun Microsystems JCP Expert Group “Mobile Information Device Profile Specifications for Java™ 2 Micro Edition Version 2.0”

List of Publication

1. Varinder pal Singh, Parteek Bhatia, “Mobile Services for Punjabi Language”, National Conference on Impact of IT on Society Emerging Trends and Issues February 28- March 2009 at DAV College Amritsar.

Appendix

The Text file that is used as back end for translation of English language to Punjabi language is attached here. The file is constructed by searching form variously existing English to Punjabi translating, transliterating application. Mainly used web resources are www.punjabonline.com, www.ijunoon.com, <http://www.punjabikhoj.com/>, <http://www.akhar.org>. This file has few thousand file and the snapshot of the file is as follows

I	Main
He	Oh
She	Oh
You	Tu, tusi
It	Eh
This	Ah
That	Oh
A	Ik
Come(you come)	Aa ja
Came	Ayea
Will come	Aooga
Open(to open)	Khollna
Opened	Kholliya
Will open	Kholluga
Sit(to sit)	Baithna
Walk(to walk)	Tuurna
Eat(to eat)	Khana
Drink	Pinna
Win	Jittna
Go	Janna
Run	Bhujjna
I go	Main jaanda haan
He goes	Oh jaanda hai
He eats an apple	Oh ik seb khanda hai
He is eating an apple	Oh ik seb kha reha hai
He ate an apple	Usne ik seb khadha
Water	PAANI
She came by bus yesterday	Oh kal bus ton aai
They went to the mosque	Oh masjid gaie
He slept the whole night	Oh saari rat sutta reha

Bring me a cup of tea
He has eaten
He had eaten
He had gone
He had come
He will eat
He will go
He will come
What is your name?
What
Is
Your
Name
What did you do?
What should I do?
What can I do?
What are the questions?
What were the questions?
What is the last question?
What is written in the letter?
What you had been told?
What will be the answer?
Why did you come?
Why did you sleep?
Why did she pay the money?
Why did they sit there?
Why do you drive the car?
How did you come?
How did you sleep?
How did you drive?
How did you write?
How many did you take?
How much did he pay you?
How much distance to go?

Mere layee chaa da cup liyao
Usne kha liya hai
Usne kha liya si
Oh chal giya si
Oh a gaya si
Oh kha lawega
Oh chala jawega
Oh aa jayeega
Tuhada naan ki hai?
Kee
Hai
Tuhadaa
Naan
Tussi ki kitta?
Tuhanu ki karna chaida hai?
Main ki kar sakda haan?
Swaal ki hann?
Swaal ki sann?
Akhri swaal ki si?
Chittthi wich ki likheya hai?
Tuhannu ki dasseyaya gaya si?
Jawaab ki howega?
Tussi kyon aaye?
Tussi soun kyon gayye?
Usne paise kyon ditte?
Oh utthe kyon baitthe?
Tussi kyon car challande ho?
Tussi kiwein aaye?
Tussi kiwein soye?
Tussi kidda chlayee?
Tussi kidda likhyea?
Tussi kinne laye?
Usne tuhanu kinna ditta?
Hor kinni door hai?