Mobile Application and E-Classes for Increasing the Availability of Information for Visually Impaired Persons

Deepak Kumar and Mohammed Abdul Qadeer

Abstract - The aim of this paper is to describe a software application that converts Mobile SMSs into Braille script with special emphasis on special symbols so that visually impaired people will also be able to read mobile messages and we have also introduced the concept of e-classes for blinds which increases the availability of information for them. At the time of receiving incoming text messages, this application allows user to select the option to convert these text messages into the Braille and then it scans these text messages, converts these messages’ text into Braille script using ASCII-Braille encoding and sends them to the electronic Braille reader or Braille Embosser attached via USB port or Bluetooth so that it can be either embossed on paper or can be read directly by Electronic Braille Reader. This application is implemented using J2ME which makes use of it generic because today most of the mobile phones are java enabled.

This application also maintains the mobility because devices used here are portable without any role of PC. For measuring the accuracy of this software, we have tested it on over about 100 messages and it provided on an average 90% accurate result when messages do not contain any multimedia content.

Index Terms - ASCII-Braille Encoding, Braille, Braille Transcription, E-class for blinds, Mobile SMS

I. INTRODUCTION

At the end of the march 2006 there were people in UK who were registered as severely sight impaired (blind) in the department of health. At 31 March 2008, 153,000 people were on the register of blind people, a slight increase of around 500 (0.3%) from March 2006[15]. In India, 10,634,000 people are blinds in which 5,732,000 are male and 4,902,000 are female. A large group of people have also significant sight loss that does not fall in this narrow category. So According to this scenario there is strong need of technological development in this field. Mobile SMS that is Short text Messaging Service is becoming very popular now-a-days since it provides a cheap means of communication. Today an average of 60% of mobile communication is done by means of SMSs. But how a blind person is going to use this service. So to resolve this problem we have implemented SMS to Braille transcription software that alters on the receiving of incoming text messages and then transcribe them into the Braille if desired and store this message into the mobile phone’s database so that they can also be used later. Then these messages in the Braille format can be transferred to the devices like Electronic Braille Reader which is used for reading Braille by means of creating Braille dot patterns through switches, and Braille Embossers which is used for Embossing Braille on to the paper through USB port or Bluetooth. This is also useful even if we don’t have installed this application on our mobile. Since it converts the incoming messages as well as the message that we type in text format into the Braille and saves them in a database. These Messages in Braille format can be sent to another visually impaired person’s mobile so that he will also be able to read that messages even if he has not installed that application because he is already receiving message in Braille format. But the limitation in this case is that his native SMS application should have the functionality of printing messages through USB port or Bluetooth. There may be some software that convert text message into speech but in case of special symbols and notation they fail and they are also language particular. They also fail when any person is blind as well as not have the capability of hearing. But our application SBT functions properly in all these cases.

II. BACKGROUND

A. A BRIEF VIEW OF BRAILLE

Braille is a system of encodings of print in embossed dot patterns used for reading and writing by the blind [1]. Each Braille character occupies a cell of fixed size. It consists of two columns of dots, numbered 1, 2, 3 and 4, 5, 6 from top to bottom. (There are also Braille codes using two columns with four dots [8, 9].)
In this paper, Braille characters are represented by dot images; as examples we list the Braille representations of the first ten characters of the alphabet which, when preceded by the indicator for numbers, also denote the ten digits.

\[
\begin{array}{cccccccccccc}
& a & b & c & d & e & f & g & h & i & j \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 0 \\
\end{array}
\]

There are \(2^{6} = 64\) Braille characters altogether with used as the space character. The dimensions of the Braille cell, according to the Library of Congress standards, are given in [5]. Certain Braille printers also permit a graphics mode. However, in this paper we do not consider Braille graphics at all. Given the fact that only 64 Braille characters are available, it is clear that special encoding rules had to be developed for different applications [3]. Different languages use different encodings (see [6, 7], for instance). In this paper, we restrict our attention to the text message to Braille conversion. In addition to the variation according to language, there exist different “grades” for the encoding: Grade I Braille, for instance, renders text with all details concerning punctuation, capitalization, spelling, and numerals, whereas grade II Braille employs a system of contractions and abbreviations [6, 10]. Typically, a grade II Braille text is 30% shorter than its grade I counterpart. We have employed an encoding technique that transcribes messages into grade I Braille so as to cover a wide range of symbols in Braille.

**B. SURVEY OF BRAILLE TRANSCRIPTION**

Before the introduction of computers into the Braille production process, Braille transcription worked essentially as follows: a person who had been trained in the relevant Braille codes would have a printed copy of the text in question and produce a Braille copy using a Braille writing machine. Typically, such a Braille writing machine has six keys for embossing a Braille character, one for each of the dots of the Braille cell, and a few more keys for operations like ‘space’, ‘backspace’, etc. In the case of single copy, the Braille is produced directly onto heavy stock paper. For large scale production, the Braille is embossed onto zinc plates which are then applied under pressure to the heavy stock paper [11]. The Braille version would have to be proofread and corrected; corrections could require that complete pages be re-done. Braille transcription is difficult, time intensive and costly. The training time for a transcriber is given as between 6 months and a year [11]; even longer periods are required for training in complicated codes like Nemeth Braille. A page of Braille takes about 30 minutes to produce on the average. This includes proofreading and corrections. Typically, a page of print results in about two pages of Braille. Given these conditions, it is clear that only a very small part of the printed publications can actually be transcribed. Moreover, access to less frequently demanded documents is very slow. Given the traditional set-up, browsing through scholarly journals or recent technical reports is something a blind reader can only dream of. With the introduction of computers into the transcription process certain simplifications became feasible. In the first step, the text is put into machine readable form. The task of translating the input into Braille is then left to a computer program. Several programs exist that afford the conversion from ASCII Braille to (literary) grade II Braille (see [12, 13, 14], for example). Some successful attempts at computer-aided transcription of mathematics have also been made [8]. However, to our knowledge, the automatic transcription into the Braille code for mobile SMSs is not offered by any software package up to now which we think an essential thing today and must be available for the blinds.

**III. SYSTEM DESCRIPTION**

Mobile SMSs to Braille Transcription is a software application that produces Braille for Mobile Messages. Braille applications for mobile developed previously makes the use of a computer system to convert the text into the Braille which did not provide the portability. Since mobile is a portable device so mobility means a lot here. Figure 3. shows the previously proposed system.

This application is implemented using J2ME. In particular we have used APIs for wireless messaging, database management, USB port access which maintains the mobility.
of system. This application listens for the receiving of incoming text messages when mobile receives any text message SBT displays a dialog box to ask whether you want to convert it into the Braille or not. On selecting yes by a single click SBT transcribes that text message into the Braille format and stores a separate copy of it in database which is maintained by it. This message in Braille format is then can be sent on the Braille device attached to the port. This Braille device is also portable. This application also provide the functionality of typing a message in text format, than convert it into the Braille format and then send it to the another blind person. But that person should have message printing facility. There are basically three major modules of this software application-

Receiving of Text Message

First module of this application is the receiving of incoming SMS. We have created an MIDlet which extends from the MIDlet class. Then we have used WMA (Wireless Messaging API) JSR120 to create a connection for sending and receiving the messages. To make a connection, the application obtains an object implementing the MessageConnection from the Connector class by providing an URL connection string that identifies the address. The code depicted below shows the receiving and sending of SMS.

```java
public boolean connectSMSServer()
{
    try {
        messageConnection = (MessageConnection) Connector.open("sms://:" + port);
        messageConnection.setMessageListener(this);
    } catch (Exception e) {  }
    public void receiveTextMessage()
    {
        try {
            Message message = messageConnection.receive();
            if (message instanceof TextMessage)
                TextMessage textMessage = (TextMessage) message;
        } catch (Exception e) {  }
    }
    public synchronized void notifyIncomingMessage(MessageConnection conn)
    {
        synchronized (this)
        { notify();  }
    }
}
```

By the code given above SBT detects the reception of incoming message. Then it displays a dialog box which will remain on the screen until user responds to it, asking whether to convert this text message into Braille or not. If user select yes then SBT performs ASCII to Braille conversion because text message received is in the ASCII form.

A. ASCII to Braille Conversion

The text message in the input phase is expected to contain only plain ASCII text or the ASCII equivalents of special Braille notation.

- Since complete information about the appearance of the printed text is present in the input text message, no information needs to be added for the transcription process.
- Correctness or rather consistency of the Braille transcription is guaranteed.
- The Braille production of a text is nearly as fast as the normal printing because the time-consuming input phase is eliminated from the process.
- Braille copies of a text messages can be produced on demand and on short notice.
- Fast and reliable communication between sighted and blind individuals because it is based on a common representation of both the printed and the Braille version of the text.

So considering all the facts mentioned above, to convert ASCII characters into the Braille characters we have developed an encoding schema, known as ASCII-Braille encoding that encodes Braille. According to this encoding schema six dots of a Braille character are considered as six bits of Braille character. Raised dots are considered as bit ‘1’ or ‘high’ and lowered dots are considered as bit ‘0’ or ‘low’ as shown in following figure.

A. ASCII to Braille Conversion

The crucial problem in computerized Braille transcription is that of obtaining the text in machine readable form with enough detail. In addition to the actual text, the required information includes certain layout specifications similar to those outlined in[15], and linearized encodings of non-linear structures and special symbols, like mathematical formulae. In both respects, it is of course not necessary that the input file conform to some specific Braille coding rules as long as a translation respecting those can be achieved. So to achieve a correct a conversion we have first collected some points.

A. ASCII to Braille Conversion

The crucial problem in computerized Braille transcription is that of obtaining the text in machine readable form with enough detail. In addition to the actual text, the required information includes certain layout specifications similar to those outlined in[15], and linearized encodings of non-linear structures and special symbols, like mathematical formulae. In both respects, it is of course not necessary that the input file conform to some specific Braille coding rules as long as a translation respecting those can be achieved. So to achieve a correct a conversion we have first collected some points.
and dot number 1,6 are lowered and considered as bit ‘0’. So In ASCII ‘Y’ is represented as 01111001 and its corresponding Braille Value is XX111010 in which two MSBs are “don’t care” and only six LSBs are significant. Now we use the ASCII characters that correspond to the converted Braille character. Using this encoding we have created a look up table in applications database. Our program reads the text message one by one character and corresponding Braille notation is fetched from the look up table and then this message is displayed on the screen in Braille Format.

<table>
<thead>
<tr>
<th>Input</th>
<th>Input Binary</th>
<th>Output Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>01110010</td>
<td>XX010111</td>
</tr>
<tr>
<td>S</td>
<td>01110011</td>
<td>XX001110</td>
</tr>
<tr>
<td>T</td>
<td>01110100</td>
<td>XX011110</td>
</tr>
<tr>
<td>U</td>
<td>01110101</td>
<td>XX100101</td>
</tr>
<tr>
<td>V</td>
<td>01110110</td>
<td>XX100111</td>
</tr>
<tr>
<td>W</td>
<td>01111011</td>
<td>XX111010</td>
</tr>
<tr>
<td>X</td>
<td>01111000</td>
<td>XX101101</td>
</tr>
<tr>
<td>Y</td>
<td>01111001</td>
<td>XX111101</td>
</tr>
<tr>
<td>Z</td>
<td>01111100</td>
<td>XX110101</td>
</tr>
<tr>
<td>1</td>
<td>00110001</td>
<td>XX000010</td>
</tr>
<tr>
<td>2</td>
<td>00110010</td>
<td>XX000110</td>
</tr>
<tr>
<td>3</td>
<td>00110011</td>
<td>XX010010</td>
</tr>
<tr>
<td>4</td>
<td>00110100</td>
<td>XX110010</td>
</tr>
<tr>
<td>5</td>
<td>00110101</td>
<td>XX100010</td>
</tr>
<tr>
<td>6</td>
<td>00110110</td>
<td>XX010110</td>
</tr>
<tr>
<td>7</td>
<td>00110111</td>
<td>XX110110</td>
</tr>
<tr>
<td>8</td>
<td>00111000</td>
<td>XX100110</td>
</tr>
<tr>
<td>9</td>
<td>00111100</td>
<td>XX110110</td>
</tr>
<tr>
<td>0</td>
<td>00111000</td>
<td>XX110010</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input (Sp. Symbols)</th>
<th>Input Binary</th>
<th>Output Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td>Semicolon</td>
<td>00111011</td>
<td>XX000110</td>
</tr>
<tr>
<td>Colon</td>
<td>00111110</td>
<td>XX010010</td>
</tr>
<tr>
<td>Exclamation</td>
<td>00111001</td>
<td>XX011011</td>
</tr>
<tr>
<td>Open Bracket</td>
<td>00101000</td>
<td>XX111010</td>
</tr>
<tr>
<td>Close Bracket</td>
<td>00101010</td>
<td>XX111010</td>
</tr>
<tr>
<td>Open Quotation</td>
<td>00100010</td>
<td>XX100110</td>
</tr>
<tr>
<td>Close Quotation</td>
<td>00100010</td>
<td>XX110100</td>
</tr>
<tr>
<td>Single Quotation</td>
<td>00100011</td>
<td>XX000010</td>
</tr>
<tr>
<td>Full Stop</td>
<td>00101110</td>
<td>XX110010</td>
</tr>
<tr>
<td>Hyphen</td>
<td>00101101</td>
<td>XX100100</td>
</tr>
<tr>
<td>Question Mark</td>
<td>00111111</td>
<td>XX110010</td>
</tr>
</tbody>
</table>

B. Accessing USB Port

Braille Output devices can be connected to the computer system through one of these four ways 1- Parallel Port 2- RS232 3- USB Port 4- Bluetooth. The last two options are feasible to provide communication between Braille device and mobile device. We have implemented a module to send a message in Braille to the serial USB port so that this message can be embossed on paper by Braille embossers or can be Read by Refreshable Braille Display connected to the USB port. JSR80 is the Java Specification Request concerned with communication with Universal Serial Bus (USB) devices. The javax.usb API matches the physical device topology, as shown in figure 7. However the root of the topology is slightly different. First, the entry point of javax.usb is the UsbHostManager class, shown at the top of the tree. This class instantiates the platform-specific instance of the UsbServices interface. From the UsbServices instance, the virtual root UsbHub is available. This hub is created by the implementation and does nothing except manage the actual physical devices. Each hub connected to the virtual root hub represents a physical Host Controller hub that is present in the system. Connected to those hubs are the real externally-connected physical devices connected to the system. Using these APIs we have created interface to send data on the devices through USB programmatically.
provides the class DiscoveryAgent for performing Device Discovery and Service Discovery. The class LocalDevice represents the local bluetooth device. The LocalDevice class provides a factory method ‘getDiscoveryAgent’ that returns a singleton instance of the DiscoveryAgent. This instance can then be used to discover other bluetooth devices and services. It first searches for all available Braille Bluetooth Devices, and prompts to select a device on which to search the OBEX Push Service. LocalDevice class can be considered as the basic entry point to the JSR-82 API. It is used to initialize the JSR-82 stack, and to control the local bluetooth settings. Each bluetooth device has a 48-bit or 6-byte value which is termed as the device’s bluetooth address. This is the unique value that differentiates one bluetooth device from the other. The address looks somewhat like this. 00:66:43:B4:67:89. Each bluetooth device has a device class or (Class of Device, CoD as specified by the bluetooth spec). This field stores information about the type of device and which types of service are supported by the device. This bluetooth device has a service database in which it can register services so as to allow other bluetooth devices to use them. The details of each registered service are kept as an object of class ServiceRecord. Each service record is associated with a Notifier object that registers it. There are notifiers for L2CAP, SPP and GOEP. To retrieve a service record corresponding to a service, the notifier that registered the record is necessary. Once the service record is fetched from the local service database it can be updated with the new values and the modified record can be added to the service database. In this way we have made a connection and access the bluetooth enabled Braille devices with our software application SBT.

### D. Screen Shots of Software

![Figure 8: Emulator showing message in Text format](image)

![Figure 9: Emulator showing message in Braille format](image)

![Figure 10: Emulator showing menus in SBT](image)

### IV. PRACTICAL APPLICATIONS

The practical applications of this software include reading and writing of SMSs for the Blinds through devices for the blind persons. The device used by the blinds for reading purpose is known as “Refreshable Braille Display” and the device used for writing Braille codes or to print out the Braille coded characters is known as “Braille Embossers”. A refreshable Braille display is a device that allows a blind person to read the contents of a display one text line at a time.
in the form of a line of Braille characters. Each Braille character consists of six movable pins in a rectangular array. The pins can rise and fall depending on the electrical signals they receive. This simulates the effect of the raised dots of Braille impressed on paper. There are usually 40, 65, or 80 arrays (characters) per line of text, depending on the device [4]. When currents or voltages are applied to points in each six-pin array, various combinations of elevated and retracted pins produce the effect of raised dots or dot-absences in paper Braille. In the piezoelectric display, each pin is mounted above a piezoelectric crystal with metal attached to one side. If a sufficient voltage is applied to the crystal, it becomes slightly shorter. This causes the metal to bow upwards, raising the pin. Thus when there is no voltage, the pin is retracted, corresponding to the absence of a dot in Braille; when there is voltage across the crystal, the pin is elevated, corresponding to a dot.

![Figure 11: Refreshable Braille Display](image)

Braille Embosser is a printer which is used for printing Braille. It embosses Braille Character on the sheet of paper with relative ease. Some Braille Embossers are Romeo Attaché, Tiger Braille Embosser, Juliet Classic Braille and Braille Express 150 and many more.

![Figure 12: Braille Embosser](image)

V. WORK UNDER PROGRESS AND FUTURE WORK

The system for SMS to Braille transcription described in the previous section can be also implemented on the Android mobiles. Generally mobile phones are largely closed environments built on proprietary operating systems that required proprietary development tools. But Android is an open source software stack backed by more than 40 Open Handset Alliances (OHA) members and is surrounded by significant industry buzz, so we can use databases and other native tools for application implementation, for this reason we have also implemented SMS to Braille transcription on Android phones. We have created ASCII-Braille encoding lookup table in SQLite database provided in Android. Whenever we receive a SMS, our Braille transcription software is alerted by the broadcast receiver and asks whether to convert it into the Braille or not. If we click for Braille transcription then message is converted into Braille by same method as described in section III (ASCII-Braille encoding table). We have provided two methods for accessing the Braille output device either by USB or Bluetooth connectivity. Today mostly all the Braille output devices are accessible by USB port and Bluetooth. In this way Android mobile can have interaction with Braille output devices by which a blind person can either read the message or emboss it on paper. We are now planning to implement e-class rooms for the blinds. In traditional class rooms for the blinds when a teacher is giving lecture it is not easy to understand all the things by listening specially in case of mathematical derivations and scientific terms. If blind students have book in Braille format then the scope of study is limited to only that book. But the concept of e-class rooms provides so much ease and overcome all limitations discussed. In e-class room there will be a computer system in front of the teacher and Bluetooth enable Braille reader in front of student. Computer has Braille transcription software installed on it. Now whatever teacher type in that software, will be sent to the Braille output devices of all the students and they can study and understand the things in more convenient way. Our work is to implement the Braille transcription software that will produce grade I Braille including mathematical and scientific symbol as more as possible, providing convenient way and tools for typing mathematical formulas and derivations and provide interaction with Braille output devices. In our Braille transcription software we will also provide web page and e-mails transcription into Braille so that students will be connected to the world. We hope this work will be very beneficial for visually impaired persons.

VI. RESULT AND CONCLUSION

We are here converting text SMS to Braille and simulating Braille output devices through attached hardware. We have tested results for over hundred SMSs of various types.

<table>
<thead>
<tr>
<th>Types of SMS</th>
<th>Number of SMS</th>
<th>Expected Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Simple text</td>
<td>30</td>
<td>100%</td>
</tr>
<tr>
<td>Text with special symbols</td>
<td>30</td>
<td>97%</td>
</tr>
<tr>
<td>Text with picture</td>
<td>30</td>
<td>75%</td>
</tr>
<tr>
<td>MMS</td>
<td>30</td>
<td>Fails</td>
</tr>
</tbody>
</table>

As shown in table for text messages results is 100% correct but since we are not considering graphics at all so result for
text message with picture depends on ratio of text and picture part in the message if text part is more than picture part then accuracy will be high, our aggregate result in such case is 75 % and in case of MMS this software fails. There are some Braille transcription software available that read word documents, text files, HTML pages and e-mails. As per the referred articles there is no Braille transcription software for Mobile SMSs. So we have here introduced the Braille transcription of Mobile SMSs with relatively easy and fast conversion method. We have tried to greater extent to produce a perfect Braille through this software. This is fully practical based application which will help the blinds for the usage of technology (SMSs). This application increases the availability of information for blinds. It also provides reliability since some mobile speech software available but what about when these messages are deleted. So Using “Mobile SMS to Braille Transcription” we can make records of messages and can use them whenever required. We hope that this software and the e-class room idea will be very helpful for the blinds.

REFERENCES


Deepak Kumar was born in India in June 1988. He has completed polytechnic in computer engineering from University Polytechnic, Aligarh Muslim University, Aligarh India in 2007. Currently he is pursuing a graduate degree in Computer Engineering (class of 2011). His area of interest includes technological growth for humanity, heterogeneous computing and WMA.

Mr. Kumar has two ACM, four IEEE and two international Publication.

Mohammed A Qadeer (MIEEE, LMIETE, LMCSI) is an Asst. Professor with the Department of Computer Engineering, Aligarh Muslim University, India. Earlier, he was working with Cisco Systems Inc. as a Network Consulting Engineer with the Advanced Services division in the APAC region. He received his B.Sc. Engineering (Computer Engineering) from Aligarh Muslim University in 1996. He has an extensive experience in the area of computer networks, wireless networks, mobile computing, next generation networks, IMS, LTE, WiMAX, 4G, WiBro etc.

Established global and nationwide setups of Internet Service Providers (ISP), Internet Exchange Points (IXP), Internet Data Centre (IDC) and Content Delivery Networks (CDN) both from a Networks and Systems perspective. His areas of research are computer networks, wireless networks, mobile computing, next generation networks, IMS, LTE, WiMAX, 4G, WiBro etc.