Bangla Text Input and Rendering Support for Short Message Service on Mobile Devices

Tofazzal Rownok, Md. Zahurul Islam† and Mumit Khan†
Department of Computer Science and Engineering, BRAC University, Dhaka, Bangladesh
† *Center for Research on Bangla Language Processing, BRAC University, Dhaka, Bangladesh*
rownok@gmail.com, zahurul@bracu.ac.bd, mumit@bracu.ac.bd

Abstract

Technology is the most important thing that involve in our everyday life. It is involving in almost every aspect of life like communication, work, shopping, recreation etc. Communication through mobile devices is the most effective and easy way now a day. It is faster, easier and you can communicate whenever you want from any-where. Mobile messaging or short message service is one of the popular ways to communicate using mobile devices. It is a big challenge to write and display Bangla characters on mobile devices. In this paper, we describe a Bangla text input method and rendering support on mobile devices for short message service.

1. Introduction

There are 14 million mobile users in Bangladesh and it is only 10% of our total population. Mobile users are increasing rapidly but problem is more than 60% of them are unable to operate their mobile properly and for this they are unable to use all the features that are included in their mobile phones.

Mobile messaging or short message service (SMS) [1-2] is one of the popular ways to communicate. It is instant and also we can send message from Bangladesh to al-most all other countries. There is no language preference to send or receive messages. So users who are unable to understand English they have the facility of sending messages but they are unable to use it. Language preference for those people may increase the usability of SMS.

To develop such application we don't have so many choices. So what we can do we can use more platforms like we can use Java 2 Micro Edition (J2ME) [3] for java enable mobile phones and Microsoft Visual Studio.Net(C#) for mobile phone based on Windows CE (Microsoft Windows for mobile and handheld devices) [4]. There are very few mobile phone users in Bangladesh, which has mobile devices with java or windows CE; the number is increasing day by day. So, writing SMS using Bangla and displayable it to users when they are typing and also displayable it to receiver is important to make the SMS facility usable to those people who are unable to understand English.

In section II, we review some related work including work on Bangla SMS, followed by our methodology in section III. We describe out design in section IV and implementation and testing details in sections V and then limitation and future improvements in section VI and finally conclude in section VII.

2. Related work

There is a similar application called 3SM System [5-6] is available for City Cell mobile phone user. In this application, there is no Unicode [7] support. They use text to image conversion procedure to send SMS. User through their mobile phone write SMS in Bangla using English letters like if some one want to write 'আমি' they write 'Ami' then send it to specific phone number then there is a server side text to image conversion they convert English Ami to Bangla আমি and make an image which is rendered only by Nokia phone. After that server side procedure they send it to specific phone number now if that specific number user using Nokia mobile no then only they can see the SMS correctly.

Another mobile phone operator, Aktel also provides Bangla SMS service [8]. This is the most advanced and user friendly version till now. It can send Bangla SMS and also capable of showing user what they are writing. This application is also device specific only java enable phone can use this application. Also both sender and receiver must have this application installed in their phones. Grameen Phone, the leading mobile phone company in Bangladesh also provides Bangla SMS service with some limitations.

3. Methodology

To develop such application for mobile devices there are some limitations like there is no standard Bangla key board layout which we all can follow. For this problem every one trying to follow their own key board layout which is disturbing for user they have to remember all of them. Also rendering [9] support for Bangla script is not available to all mobile devices and it is depend on manufacturer. So we need standardization. Also to develop such application we can not use any programming language.

The challenges in adding Bangla support to SMS include an input method for Bangla as well as a rendering system capable of rendering the Bangla script on the mobile display. The input method can be

implemented by creating a key map. The rendering system is much more complex as most mobile devices do not include support for rendering Indic scripts such as Bangla.

3.1. Text input method

We are using a data structure where there will be an assigned English character for each Bangla Unicode character. Like when user presses a button to write অ (first Bangla letter) what user is pressing English letter but we will come up with corresponding Bangla letter. For mapping all the characters we have followed a keyboard layout. Avro keyboard [10] is a well established Bangla Input method for computer. Figure 1 shows the Key map technique for Bangla text input.

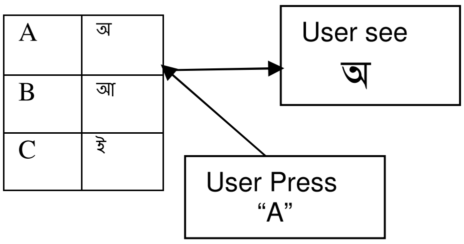

Figure 1: Key mapping technique

3.2. Rendering

Now to display these Bangla Unicode characters [7] to user in their mobile device's screen we have to render the texts. Rendering means process of generating text Bangla is an Indic script. There are some rules, features of Indic script to render it in a screen.

There are vowels and consonants, vowels are written in specific position relative to consonant like- ক া → কা it is knows as reordering.

Sometimes in one vowel we have two components so according to consonants we have to place these components in a way so that it take place appropriately with consonants like- ক ো → কো

Bangla Language has few ligatures and these are most important features of Bangla language. Some times we have to combine multiple characters into a single character. In contextual shaping the shape of a character is depends on its neighbor characters or sometimes position within the word.

Then we have diacritic placement it is about smart positioning there are few characters which placed together like- ু ূ. So how will it shown in mobile screen. Now to render text with these indic script features we generated some rules so that we can show specific bangla characters in mobile devices.

4. System design

The requirements of Bangla SMS systems are type message in Bangla and display Bangla text in receiver end. Figure 2 shows the use case diagram of the system. We are describing our system using full dressed use case. Figure 3a and 3b shows system sequence diagram and Figure 4 shows the model view controller of the system.

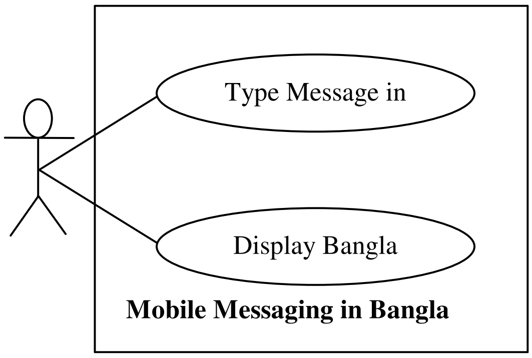

Figure 2: Use case diagram of the system

4.1. Use case: Type message in Bangla

`Primary Actors`: User.
`Stakeholders`: User – Wants to write message in Bangla.
`Precondition:`

- User must have knowledge of how to write Bangla.
- Input character within 160 characters.

`Post condition:`

Characters are correctly formatted.

`Main Success scenario:`

1. User will open the menu from his/her mobile.
2. Then choose write message option from menu.
3. An editor will open to type message.
4. To write words user will enter some letters from mobile keypad.
5. System will take the letter given by the user and come up with the specific Bangla letter within the information system has.

`Extensions:`

*a. At any time system fails
1. Restart the system.

4a. Typing Error:
1. System signals error and reject entry.

`Special Requirements:`

None

`Technology and Data Variations List:`

- Mobile keypad will use to write the text.

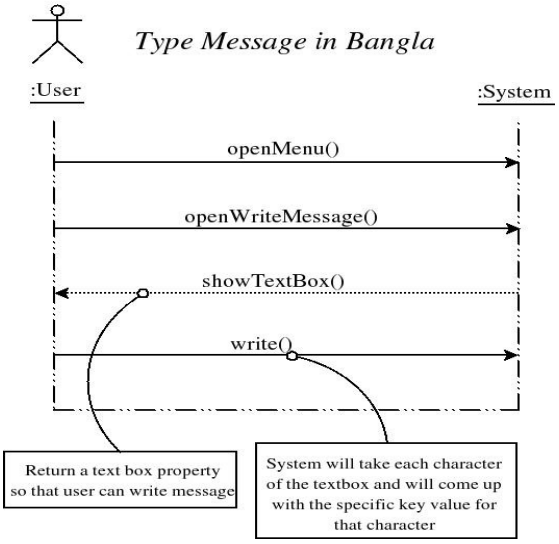

Figure 3a: System Sequence diagram of the use case -Type message in Bangla.

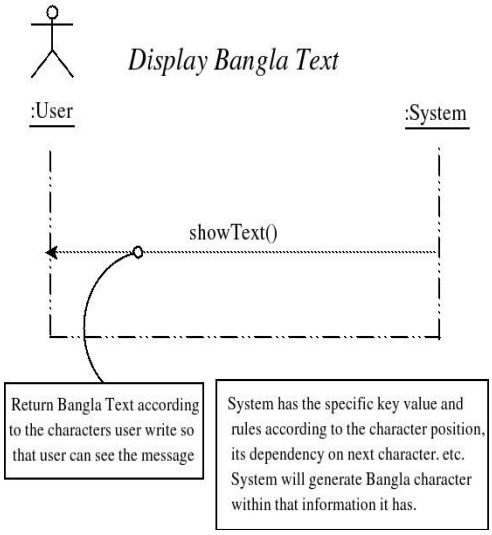

Figure 3b: System Sequence diagram of the use case - Display Bangla text

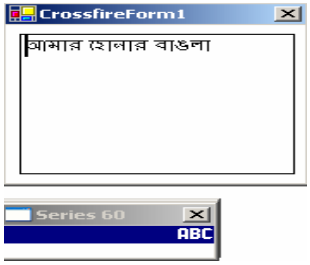

Figure 4a: Bangla text on Appforge Crossfire

Figure 4b: Bangla text on .NET IDE

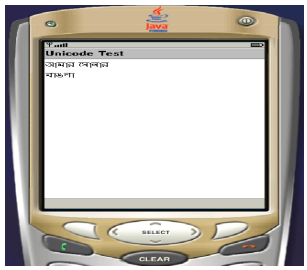

Figure 4c: Bangla text on J2ME emulator

4.2. Use case: Display Bangla text

`Primary Actors`: User.
`Stakeholders:`
User – Wants to see what he/she is typing.
`Precondition:`
- User must type something to display.

`Post condition:`
- Display text as written.

`Main Success scenario:`
1. System has the specific information and key values.
2. System check the rule within this key value about its positioning or other features like- is it dependent on next character or not, splitting etc..
3. After analysis all rules related with that key value system has to come up with a glyph data or the actual Bangla character to show.
4. Then show the character according to rules and its key value.

`Extensions:`
*a. At any time system fails
1. Restart the system.
4a. Error in Displaying:
1. Find out the error is it problem with positioning or anything else then solve it according to the problem.

`Special Requirements:`
None
`Technology and Data Variations List:`
Indic rendering rules will use to display the text.

4.3. Model view controller

Model – Data representation.
View – User Interfaces.
Controller – Event Handling.

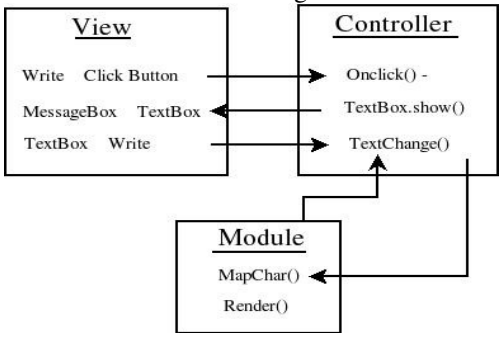

Figure 5: Model View Controller of the system

5. Implementation and testing

To implement this application we have used J2ME [3] and C#. J2ME for mobile phones which are using Symbian OS [11] and other mobile phones, which are not using Symbian OS, but Java enable. C# is for mobile phones which are using Windows CE and other mobile phones which are using Palm OS. We use few classes for both platforms and these are:

- BanglaUi
- BanglaController
- BanglaModel

5.1. BanglaUi

```
public class BanglaUi extends MIDlet  {
   Display display;     // Object of Display
Class to display the form.
   Form form = null; // Object of Form Class
where all displayable object
like-text field etc will take place.
   TextField msg = null; // Object of Text
Field Class where user will write.
   BanglaControl  send=null;  //  Object  of
BanglaController Class which will
handle events of user interface.
   public BanglaUi() {
   }
   public void startApp() {
      // after starting the application what
will given to user will add here.
   }
   public void pauseApp() {
     // If there is any interrupt from the
device what will the application do will add
here.
   }
   public       void       destroyApp(boolean
unconditional) {
     // after user close the application if
anything to do ,    will add here.
   }
}
```

5.2. BanglaController

```
public class BanglaControl  {
   BanglaModel  mapper=new  BanglaModel();  //
Object of BanglaModel Class.
   BanglaUi  message=null;      //  Object  of
BanglaUi Class.
   Char text_char [];
   public  BanglaController(TextField a){
       //Text  field  Item  will  be  come  from
the  BanglaUi  and  Item  State      Listener
Interface  is used for change the text .
      ItemStateListener   listener   =   new
ItemStateListener() {
           public  void  itemStateChanged(Item
item) {
              // Text field Item will be come
from the BanglaUi
             TextField  f  =  (TextField)  item;

            f.getChars(text_char);        //
Copies the text field contents to Character
array.     }
       };
       message.form.setItemStateListener(list
ener);// set event handler for the BanglaUi
Class
    }
```

5.3. BanglaModel

```
public class BanglaModel {
   char vowels[ ][ ];
   char consonants[ ][ ];// for holding the
English characters and corresponding Bangla
Character.
    Public void CharMap(char keys[ ]){
         //  It will get the every character
of the text field from controller and then
check the character in the character array it
has (Vowels , Consonants)and will give the
corresponding bangla character .
        public BanglaModel ()
     }
}
```

To test this application we deployed in emulators to see the desired output. There are several emulator based on Symbian OS Nokia Series 60 [11], Microsoft windows CE and J2ME. We used *Series 60 2nd Edition SDK for Symbian OS, Supporting Feature Pack 3.* We also used Appforge Crossfire [12] which create environment same as .NET but it have some Symbian OS specific features and libraries. Figure 4a shows Bangla text on Appforge Crossfire. We used .NET IDE (Integrated development Environment) because of debugging facility available for application that is written in C#. Figure 4b shows Bangla text in .NET IDE for Windows CE. Also figure 4c shows Bangla text on J2ME emulator.

6. Limitations and future improvements

The application will works only mobile devices based on Symbian OS, Microsoft Windows CE or Java enable. So there is a restriction for users. Feature like-rendering Indic script is available on only new and next generation mobile devices. So to get full benefit we need to be update with technological improvements.

The application does not include any intelligent input system or spell checking. Writing Bangla in a mobile device is much more complicated then writing in a computer. Because to find a desire character we need to press a key several times also we have ligatures. So in future an intelligent input system like T9 [13] can add with the application so that user can find desire word more easily. It will reduce key pressing and also time to write a SMS.

We also need to concentrate to the receiver end of SMS. Because in receiver end there is also need an application to view the content of the SMS that is Bangla. To implement such application we can use character to image or bitmap conversion logic for mobile devices that don't have our requirements. If we can use character to image or bitmap convert processing to send SMS then receiver may able to see the content from mobile devices, which support image or bitmap.

7. Conclusion

One of the Challenges of send and receive Bangla SMS is that mobile phones with support of rendering Indic script, Java etc. has high price. Most of the mobile phone users in Bangladesh are unable to buy a mobile phone that has such features. For this such application may not a good solution but as technological improvement is increasing and everybody is looking forward we also have to cope up with improved technology to be beneficial. This application also can be integrated by mobile providers through which deployment process can be omitted.

8. References

[1] C. Peersman, S. Cvetkovic, P. Griffiths, H. Spear, "The Global System for Mobile Communications Short Message Service", in *Personal Communications, IEEE*, 2000, page: 15-23.

[2] C. Peersman, H. Spear, S. Cvetkovic, C. Smythe, "A tutorial overview of the short message service within GSM", in *Computing & Control Engineering Journal*, 2000, page: 79-89.

[3] J. White, D.A. Hemphill, *Java2 Micro Edition*, Manning Publications Co, 2002.

[4] J. Murray, *Inside Microsoft Windows CE*, Microsoft Press Redmond, WA, USA, 1998.

[5] 3SM SMS system documentation available at: http://www.banglasms.com/

[6] CITYCELL SMS system documentation available at: http://www.citycell.com/banglasms.php

[7] J. Bettels, F.A. Bishop, "Unicode: A Universal Character Code", *Digital Technical Journal*, 1993.

[8] Documentation about Aktel Bangla SMS available at: http://www.aktel.com/s_offer_Bangla_SMS.php

[9] S. Correll, "Graphite: An Extensible Rendering Engine for Complex Writing Systems", In the *proceeding of 17th International Unicode Conference* 1 San Jose, California, 2000.

[10] Avro Keyboard Layout documentation available at: http://www.omicronlab.com/avrokeyboard/

[11] M. de Jode, *Programming Java 2 Micro Edition for Symbian OS: A developer's guide to MIDP 2.0*, Symbian Press.

[12] Appforge official website: www.appforge.com

[13] M.D. Dunlop and A. Crossan, "Predictive text entry methods for mobile phones", *Personal Technologies*, 2000, pp. 134-143.