CONTENTS

1. ABSTRACT

This invention has been developed to overcome the existing barriers for non-English phonetic languages, specifically Sinhala and Tamil languages, in the field of IT. The invention is in the form of an Interface that could be interfaced with the English keyboards without the Unicode while keeping the exclusivity of the content. The interface consists of six algorithms for Sinhala and Tamil including sorting algorithms which can be directly pluggable to existing database systems. The product includes an implemented design for SMS/MMS applications for Sinhala and Tamil. The application provides generalized architecture to develop any kind of phonetic language applications with the existing resources in the industry.

2. BACKGROUND

Information technology has been spreading thorough out the Sri Lankan community for few decades. The requirement for applications of IT through national languages is obvious in this process.

Even though a number of Sinhala/Tamil fonts are available for computing, their usage is still limited to typists and typesetting experts due to the difficulties in keyboard entering. It is obvious that an average computer user, with this restless life style, will have no time or opportunity to practice the native keyboards. Although Unicode is now becoming available in Sinhala, the software vendors still haven't made use of it. Application of Sinhala and Tamil languages in software development has been kept in a minimum level due to that incompatibility of fonts and language.

There are several approaches taken by numerous individuals and organizations to use Sinhala in computing. Some of the available solutions can be listed as following

- Sinhala word : is a software application which enables the user to type Sinhala letters. This application is limited to word processing and only the developers font can be used to type. The typing method used requires considerable training and it is difficult to use sufficiently large documents due to the use of function keys.

- Sri-word : is also a software application to facilitate the user with Sinhala typing using phonetic sounds. User is allowed to type as they are pronounced in the Standard English keyboard, then it is converted to Sinhala. Only the provided font can be used and no flexibility due to the direct map to the keyboard inputs.

- Phonetic keyboard in kaputadotcom font : This system uses a keyboard layout where the user can easily remember commonly used characters as they are allocated to phonetic sounds.

- Sinhala picture messaging service used by- mobile service providers : All the mobile service providers in Sri Lanka use the same mechanism at present; a server application is used to send static picture messages. The subscriber can only receive an already created picture message which is limited to two or three words (Celltel provides a web interface to create pictures). At first, Subscriber must receive the message from the operator and then he can forward it. He cannot directly send the message to the recipient.

The system described here is not based on the techniques mentioned above, but it is a new concept that can be a universal solution to all the situations and languages. Following are the key features of the interface.

- Generalized architecture for all phonetic languages : The concept described here is an interface, not a single software application. This system provides a generalised concept for all phonetic languages and it has been implemented for Sinhala and Tamil. The technique used is based on writing analysis instead of making a direct map to each letter as done in previous systems. Type setting application is a single usage of that product. It also differs from existing software applications in the following areas. They are described in detail in the section 3.10

- Font Independence
- Unique map to Sinhala/Tamil letters
- Capability to handle almost all the presently using writing patterns
- Same software application can be used to type Sinhala and Tamil (or any other phonetic language) simultaneously.
- Sorting
- Dynamic SMS/MMS application
- Language translation

3. DISCLOSURE

3.1. LANGUAGE ANALYSIS

The architecture of the system can be directly applied for any phonetic language since its development is based on generalized analytical structure. The following analysis of Sinhala and Tamil languages describes the nature of the language writing, but several facts and patterns have been mentioned under the analysis part that are unique interface categorizations to be efficiently used in computing.

3.1.1. Sinhala

The Sinhala language is called as an '*Arya Bhasha*' since it has been developed with the influences of Aryan Languages in Northern India. *Pali* and *Sanskrit* were the major sources of the language. In consideration of writing, the historical facts prove that the present set of characters is derived version of ancient *Brahmi* letters. The alphabet has been changed with the time. Even very recently several characters (ෆ) were added while some are going to be obsolete in the future (ඏ ඐ). The Sinhala alphabet comprises with vowels, consonants and two special letters called *anuswaaraya* and *visrgaya*.

3.1.1.1. Vowels (ස්වර)

According to the mix-Sinhala alphabet, there are 17 vowels.

අ ආ ඇ ඈ ඉ ඊ උ ඌ ඍ ඎ එ ඒ ඓ ඖ ඔ ඕ ඖ

Some characters (ඎ ඏ ඐ) are not important in present writing. In writing, certain vowels are written as a combination of a vowel character and a stroke.
Eg: ආ ඇ ඈ ඓ ඖ

3.1.1.2. Consonants (ව්‍යංජන)

The Mix-Sinhala alphabet exhibits 38 consonants in Sinhala. Since the letters ඣ and ඞ are rarely used in writing, they are not even included in most of the Sinhala fonts.

ක	ඛ	ග	ඝ	ඞ
ච	ඡ	ජ	ඣ	ඤ
ට	ඨ	ඩ	ඪ	න
ත	ථ	ද	ධ	ණ
ප	ඵ	බ	භ	ම
ය	ර	ල	ව	
ශ	ෂ	ස	හ	ළ
ෆ	ඟ	ඳ	ඹ	

3.1.1.3. anuswaaraya and visrgaya

The Sinhala character set comprises with two special figures called *anuswaaraya* and *visrgaya*. They are generally written with the vowel අ.

(අ)ං *anuswaaraya*

(අ)ඃ *visrgaya*

3.1.1.4. Analysis.

In Sinhala, vowels themselves are letters. Consonants are combined with vowels and form letters. Before combining with a vowel, a consonant is in its original form called *hal akshara* .This state is also considered as a letter.

A consonant may combine with an original form of a consonant. This forms combined letters (බැඳි අකුරු). A special case occurs when a consonant combine with ර් and ය්. Two strokes are available in Sinhala to represent this combination.

Eg: ක් + ර් + අ = ක්‍ර

ක් + ය් + අ = ක්‍ය

In Sinhala the formation of vowel-consonant combination characters is not in a generic pattern. When analysing the language, 3 types of letters could be observed.

Type	Original form stroke	+ උ representation	Example
1	්	ු	ග, ක
2	්	ු	ව, ට
3	්	ු	ල, න

Table 3.1-1 Sinhala consonants – Type analysis

The consonant is ර is a special case. It doesn’t form a combination with *rakaaransa*(්‍ර) or *yansaya*(්‍ය). Its other combinations also differ from remaining consonants.

ර් + ඇ = රැ (Special stroke is used)

ර් + ඈ = රෑ (Special stroke is used)

ර් + උ = රු (The stroke is used to represent the ඇ combination in other consonants.)

ර් + ඌ = රූ (The stroke is used to represent the ඈ combination in other consonants.)

3.1.2. Tamil

Tamil is one of the languages that most commonly used in southern India at present. The same language is used in Sri Lanka with minor deviations. Tamils and Muslims use the Tamil as there communication medium. The vocabularies of the two nations may slightly differ. The Tamil alphabet consist of vowels, consonants and a special character called *aayutha eluththu* (ஆயுத எலுத்து).

3.1.2.1. Vowels (உயிர் எலுத்து)

The Sri Lankan Tamil alphabet presently comprises 12 vowels.

அ ஆ இ ஈ உ ஊ எ ஏ ஐ ஒ ஓ ஔ

3.1.2.2. Consonants(மெயி எலுத்து)

There are 23 consonants and one combined letter (ஸ்ரீ) *shriina* in the Tamil alphabet.

க ங ச ஞ ட ண த ந
ப ம ய ர ல ள வ ழ
ற ன ஜ ஷ ஹ க்ஷ

3.1.2.3. Special letter (∴ - ஆயுத எலுத்து)

This letter is combined with consonants and forms a letter. These letters sounds similar to the original form of Sinhala letters(හල් අකුරු)

3.1.2.4. Analysis

In Tamil, a vowel alone is a letter. Other letters are derived from combinations of vowel-consonants.

Even in Tamil, there isn't a generic method to obtain the letter. When the language is analysed, four categories could be observed.

Type	+ உ representation	+ ஊ representation
1	கு	கூ
2	து	தூ
3	பு	பூ
4	ஜு	ஜூ

Table 3.1-2 Tamil consonants – Type analysis

3.2. PHONETIC MAPPING

A phonetic map has been defined in order to manipulate the Sinhala and Tamil letters without influencing the nature and information content of the word. Letters are assigned keys in the Standard English key board to type without a special practice. This is based on a careful study of formation of Sinhala letters and words.

අ	ආ	ඇ	ඈ	ඉ	ඊ	උ	ඌ
a	aa	E	EE	i	ii	u	uu
ඔ	ඕ	එ	ඒ	ඓ	ඍ	ඎ	
o	oo	e	ee	ai	iRu	iRuu	
ක	ඛ	ග	ඝ	ඥ	ච	ඡ	
k	kh	g	gh	gn	ch	Ch	
ජ	ඤ	ට	ඨ	ඩ	ඪ	න	
j	kn	t	T	D/x	Dh/X	n	
ත	ථ	ද	ධ	ණ	ප	ඵ	
th	Th	d	dh	N	p	ph	
බ	භ	ම	ය	ර	ල	ව	
b	bh	m	y	r	l	v	
ශ	ෂ	ස	හ	ළ	ෆ		
sh	Sh	s	h	L	f		
ං	ඃ	ඟ	ඳ	ඬ	ඹ		
z/n	zz	NG/ng	ND/nd	MD/md	MB/mb		

Table 3.2-1 Phonetic map for Sinhala alphabet

Both the lower case and upper case letters has been used to fulfil the map for large number of letters in the Sinhala/Tamil alphabets. Certain consonants are mapped to two letters because typing applications require easier key map for the user.

As an example :

The unique map for the letter ඳ is 'ND'. But in the typing application, if the user type 'nd' or 'ND' it can be converted to letter ඳ. Since 'nd' is confusing with න්ද , the application will ask for the appropriate choice when the user input as 'nd'

The letter 'D' has been assigned to letter ඩ to be the native choice of inexperienced user. But it will automatically be converted to 'x' by the logic in order to make the integrity of the context since it would be confusing in a word like කණ්ඩායම (kaNxaayama).

Formation of letters with vowel- consonant, vowel-vowel-consonant combinations are assigned as following.

Formation	Formed letter	User input
ක්	ක්	k
ක් + අ	ක	ka
ක් + ආ	කා	kaa
ක් + ඇ	කැ	kE
ක් + ඈ	කෑ	kEE
ක් + ඉ	කි	ki
ක් + ඊ	කී	kii
ක් + උ	කු	ku
ක් + ඌ	කූ	kuu
ක් + එ	කෙ	ke
ක් + ඒ	කේ	kee
ක් + ඔ	කො	ko
ක් + ඕ	කෝ	koo
ක් + ඓ	කෛ	kai
ක් + ඖ	කෞ	kau
ක් + ං	කං	kz
ක් + ඃ	කඃ	kzz
ක් + ර් + අ	ක්‍ර	kra
ක් + ර් + ආ	ක්‍රා	kraa
ක් + ය් + අ	ක්‍ය	kya
ක් + ය් + ආ	ක්‍යා	kyaa

Table 3.2-2 Formation of letters - Sinhala (ක)

அ	ஆ	இ	ஈ	உ	ஊ	எ	ஏ	ஐ	ஒ	ஓ	ஔ	∴
a	aa	i	ii	u	uu	e	ee	ai	o	oo	au	A
க	ங	ச	ஞ	ட	ண	த	ந	ப	ம	ய	ர	ல
k	KN	s/ch	NG	d	N	th	x	p	m	y	r	l
ள	வ	ழ	ற	ன	ஜ	ஸ	ஷ	ஹ	க்ஷ	ஸ்ரீ		
X	v/w	L	R	n	j	S	sh	h	KS	shrii		

Table 3.2-3 Phonetic map for Tamil alphabet

Formation	Formed letter	User input
க்	க்	K
க் + அ	க	Ka
க் + ஆ	கா	Kaa
க் + இ	கி	Ki
க் + ஈ	கீ	Kii
க் + உ	கு	Ku
க் + ஊ	கூ	Kuu
க் + எ	கெ	Ke
க் + ஏ	கே	Kee
க் + ஐ	கை	Kai
க் + ஒ	கொ	Ko
க் + ஓ	கோ	Koo
க் + ஔ	கௌ	Kau

Table 3.2-4 Formation of letters - Tamil (க)

3.3. CONVERSION PRINCIPLE

The conversion is done font independently by analyzing phonetic input according to the language rules. The system doesn’t use a database technique to map the letters, but it does a run time analysis.

A technique has been implemented with a configuration file containing set of pointers to the font in order to distinguish the font key board layout. These configuration file is a simple text file having the file size of few kilobytes.

Line in the file	.sdb example	.tdb example
phonetic code list	k	k
Sinhala letters	ක	க
Letter type	2	1
list of strokes	ා	ா
list of vowel letters	අ	அ
phonetic code of special letters	vi	thu
Special characters	වි	து

Table 3.3-1 Configuration file structure

A sentence is analyzed sequentially from the first letter by splitting into vowels and consonants. When a known pattern is found, the conversion process seeks the appropriate letters and strokes from the configuration information and provides the accurate view using a user specified font.

The overall system can be summarized as

A phonetic language computer interface system, Which analyzes the phonetic language inputs according to their formation of vowel-consonant combinations and, Handle the language(phonetic) information using a conventional character set assigned based on their vowel-consonant combination where, Employs sorting algorithms based on the phonetic formation of letters by assigning weights to individual letters in order to provide a user friendly and programmer effective interface for language handling without bothering the writing conventions of the language

3.4. SINHALA CONVERSION

Input the phonetic text to be converted

Start from the first letter.

Recursive until the end of the sentences

```
↳ 1st letter = New line, space or Tab key → Append to the converted text as it is
↳ 1st letter = Number or punctuator → find the appropriate match from the font using the
  'Special characters in the configuration file and put it
↳ IF First letter is a vowel
    ↳ Second letter = First letter and Second letter ≠ Third letter → Identify first two letters as
      long vowel (Ex. aa)
    ↳ First two letters are ai, au → identify the vowels and convert
    ↳ First 3 or 4 letters for iRu,iRuu → identify the vowels and convert
    ↳ Else identify the vowels and convert
↳ IF First two letters form a sangngnaka (mb,md,nd,ng ) - (If in lowercase prompt before
  proceed)
    ↳ If the 3rd letter is a vowel
        ↳ Third letter = Fourth letter → Identify the 2nd two letters as long vowel sound
        ↳ Third and Fourth form ai, au → Identify the combination
        ↳ Else identify the single vowels sound and convert
    ↳ If third letter is 'r' → set a flag to identify rakaransaya
        ↳ 4th letter is a vowel
            ↳ 4th vowel = 5th vowel → convert the long vowel sound with rakaransaya
            ↳ 4th and 5th form ai,au → convert relevant with rakaransaya
            ↳ Else put the rakaransa combination with 4th letter vowel.
        ↳ Identify first two letters as hal akshara and convert. (No rakaransa)
    ↳ If third letter is 'y' → set a flag to identify yansaya
        ↳ 4th letter is a vowel
            ↳ 4th vowel = 5th vowel → convert the long vowel sound with yansaya
            ↳ 4th and 5th form ai,au → convert relevant with yansaya
            ↳ Else put the rakaransa combination with 4th letter vowel.
        ↳ Identify first two letters as hal akshara and convert. (No yansaya)
    ↳ Identify first two letters as a hal akshara and change the pointer to 3rd letter.
↳ If the Second letter is a vowel. (First should be a consonant)
    ↳ 2nd  letter = 3rd letter → Identify the 2nd two letters as long vowel  sound
    ↳ 2nd and 3rd form  ai, au → Identify the combination
    ↳ Else identify the single vowels sound and convert the two letters.
↳ If 2nd letter is 'h' and there is a map for first two letters in the allocation table → Identify
  Mahaprana has found
    ↳ If  the 3rd letter is a vowel
        ↳ Third letter = Fourth  letter → Identify the 2nd two letters as long vowel  sound
        ↳ Third and Fourth form  ai, au → Identify the combination
        ↳ Else identify  the single vowels sound and convert
    ↳ If third letter is 'r' → set a flag to identify rakaransaya
        ↳ 4th letter is a vowel
            ↳ 4th vowel = 5th vowel → convert the long vowel sound with rakaransaya
            ↳ 4th and 5th form ai,au → convert relevant with rakaransaya
            ↳ Else put the rakaransa combination with 4th letter vowel.
        ↳ Identify first two letters as hal akshara and convert. (No rakaransa)
    ↳ If third letter is 'y' → set a flag to identify yansaya
        ↳ 4th letter is a vowel
            ↳ 4th vowel = 5th vowel → convert the long vowel sound with yansaya
            ↳ 4th and 5th form ai,au → convert relevant with yansaya
            ↳ Else put the rakaransa combination with 4th letter vowel.
        ↳ Identify first two letters as hal akshara and convert. (No yansaya)
    ↳ Identify first two letters as a hal akshara and change the pointer to 3rd letter.
↳ If 2nd letter is ‘r’ and first letter ≠ ‘r’→ set a flag to identify rakaransaya
    ↳ 3rd letter is a vowel
        ↳ 3rd vowel = 4th vowel → convert the long vowel sound with rakaransaya
        ↳ 3rd and 4th form ai,au → convert relevant with rakaransaya
        ↳ Else put the rakaransa combination with 3rd letter vowel.
    ↳ Identify first two letters as hal akshara and convert. (No rakaransa)
↳ If 2nd letter is ‘y’ and first letter ≠ ‘r’→ set a flag to identify yansaya
    ↳ 3rd letter is a vowel
        ↳ 3rd vowel = 4th vowel → convert the long vowel sound with yansaya
        ↳ 3rd and 4th form ai,au → convert relevant with yansaya
        ↳ Else put the yansaya combination with 3rd letter vowel.
    ↳ Identify first two letters as hal akshara and convert. (No yansaya)
↳ Else identify first letter as a single consonant and convert with appropriate pil.
```

3.5. TAMIL CONVERSION

```
Input the phonetic text to be converted
Start from the first letter.
Recursive until the end of the sentences
    ↳ 1st letter = New line, space or Tab key → Put them as it is
    ↳ 1st letter = Number or punctuator → find the appropriate match from the font using the
      ‘Special characters in the configuration file and put it
    ↳ IF First letter is a vowel
        ↳ Second letter = First letter and Second letter ≠ Third letter → Identify first two letters as
          long vowel (Ex. aa)
        ↳ First two letters are ai, au → identify the vowels and convert
        ↳ Else identify the vowels and convert
    ↳ If  first five letters form ‘shrii’ → identify put shrina
    ↳ If the Second letter is a vowel. (First should be a consonant)
        ↳ 2nd  letter = 3rd letter → Identify the 2nd two letters as long vowel  sound
```

↳ 2nd and 3rd form ai, au → Identify the combination

↳ Else identify the single vowels sound and convert the two letters.

↳ If 2nd letter is ‘h’ and there is a map for first two letters in the allocation table → Identify *Mahaprana* has found

↳ If the 3rd letter is a vowel

↳ Third letter = Fourth letter → Identify the 2nd two letters as long vowel sound

↳ Third and Fourth form ai, au → Identify the combination

↳ Else identify the single vowels sound and convert

↳ Identify first two letters as a *hal akshara* and change the pointer to 3rd letter.

↳ Else identify first letter as a single consonant and convert with appropriate

3.6. BACKWARD CONVERSION

The concept of backward conversion is to decode Sinhala content appeared in Sinhala font to the phonetic notations defined above. This method can be enormously used in making graphical user interfaces in phonetic languages and other applications which require information manipulation irrespective of the font.

This can be used to convert existing documents to newly made fonts without damaging the content. As an example:

A document written using the FMDerana font contains	: විජේසේකර යතුරු පුවරුව
The backward conversion will decode to	: wijeeseekara yathuru puwaruwa
Now it can be written in any other font	: විජේසේකර යතුරු පුවරුව

This is achieved by keeping a configuration file to different fonts which contain the layout information.

Line in file	.rev Sinhala	.rev Tamil
vowel letter in Sinhala font	අ	அ
phonetic code for vowels	a	a
Special characters	වි	து
phonetic code of special letters	vi	thu
list of consonant letters	ක	டு
phonetic code list	k	D
stroke character	ා	ர்
Stroke code	4	1

Table 3.6-1 Reverse conversion file (.rev) structure

3.7. SORTING PRINCIPLES IN PHONETIC LANGUAGES

The sorting rule in phonetic languages is based on the formation of letters. It is required to identify the category of the letter and analyze its construction before deciding the alphabetical position of a character.

In Sinhala and Tamil, following rules are applied to decide sorting position.

- Vowels get priority and they are ranked according to the alphabetical position.
- Consonants obtain second priority for the original mode (*hal akshara*) according to their alphabetical order.
- Any form of a consonant which is a combination with a vowel will be ranked first by the consonant and then by its combined vowel.
- Consonant-Consonant-vowel combination will be ranked in the above order providing first priority to the first consonant.

In this system, letters are converted to phonetic codes mentioned above in order to handle them in a general manner.

3.8. SINHALA SORTING ALGORITHM

A critical issue in developing sorting algorithm is to make it compatible with existing database systems. It is also required to manipulate information in different languages in a generalized manner in developing multi-linguistic software interfaces. Since the rank of a letter is based on its formation, the evaluation system is based on the origin of a letter.

For Sinhala language, three letter codes are defined for each letter which can be uniquely ranked and positioned in the alphabetical order.

අ	ආ	ඇ	ඈ	ඉ	ඊ	උ	ඌ	එ	ඒ
b	c	d	e	f	g	h	i	j	k
ඔ	ඕ	ඓ	ඖ	ඍ	ඎ	අං	අඃ		
l	m	n	o	p	q	r	s		
ක	ඛ	ග	ඝ	ඞ	ච	ඡ	ජ	ඣ	ඤ
ab	ac	ad	ae	af	ag	ah	ai	aj	ak
ට	ඨ	ඩ	ඪ	න	ත	ථ	ද	ධ	ණ
al	am	an	ao	ap	aq	ar	as	at	au
ප	ඵ	බ	භ	ම	ය	ර	ල	ව	ශ
av	aw	ax	ay	az	ba	bb	bc	bd	be
ෂ	ස	හ	ළ	ෆ	ඟ	ඳ	ඬ	ඹ	
bf	bg	bh	bi	bj	bk	bl	m	bn	

Table 3.8-1 Value assignment for Sinhala alphabet

Further to the above allocation,

a – is added as suffix to indicate a *hal akshara*
aa - is added as a prefix to denote a vowel.

Above coding can be done to complete strings, and the coded string would describe the sorting rules of that sentence. The application then simply sort the sentences by the order of the coded strings.

Eg:
Sinhala String = මොරටුව විශ්ව විද්‍යාලය
After decoding = moratuwa vishva vidyaalaya
Sorting code generated = azlbbbalhbdb bdfbeabdb bdfasabacbcbbab

3.9. TAMIL SORTING ALGORITHM

The logic behind the Tamil sorting algorithm is almost the same. A Tamil letter can be represented by two letter code since the number of consonants is low compared to the Sinhala alphabet.

அ	ஆ	இ	ஈ	உ	ஊ	எ	ஏ	ஐ	ஒ	ஓ	ஔ	ஃ
a	b	c	d	e	f	g	h	i	j	k	l	m
க	ங	ச	ஞ	ட	ண	த	ந	ப	ம	ய	ர	ல
b	c	d	e	f	g	h	i	j	k	l	m	n
ள	வ	ழ	ற	ன	ஜ	ஸ	ஷ	ஹ	க்ஷ			
o	p	q	r	s	t	u	v	w	x			

Table 3.9-1 Value assignment for Tamil alphabet

Further to the above allocation,
a – is prefixed to indicate a vowel
m – is already allocated to indicate *hal akshara*

Eg:
Tamil String = வணக்கம்
After decoding = vanakkam
Sorting code generated = pasabmba

3.10. WORD PROCESSING APPLICATION – BHASHITHA

The software application Bhashitha is developed based on the previously described conversion methods in Sinhala and Tamil in order to provide efficient typing interface to those languages. The significant functionalities of the software can be listed as following.

- Supports for all of the available fonts.

The software facilitates the user to enable all the available fonts regardless of their character positions. The user will only need to select the font and appropriate letters in the new font.(Edit- Add New Font). It is also possible to obtain the font's configuration file from the developer through an e-mail.

- Capability to handle almost all the presently using writing patterns.

Most of the currently used writing styles of Sinhala and Tamil have been taken into account during the process of developing this conversion. All the *rakaaransa*(්‍ර) and *yansaya*(්‍ය) applications have been considered and they will be placed appropriately.

The consonant ර (ra) is a special case since it does not combine with any of the above.

Eg:
Mahaachaarya : මහාචාර්ය
adhipathyaya : අධිපත්‍යය
shrii vikrama : ශ්‍රී වික්‍රම
vaidya vidyaava : වෛද්‍ය විද්‍යාව

It is required to assign a letter for ඇ, a unique vowel to Sinhala since its pronunciation does not directly match with a letter in English alphabet. During the process, 'E' is assigned to ඇ since 'e' has been dedicated to represent එ. A new user will only need to remember this point and put two vowel letters when the pronunciation is long.

The software offers several facilities to the user. The 'Ask For Clarification' option has been established to overcome certain confusing points in this mapping process. As an example the Sinhala letter 'ං' and 'න් ' can be represented by the letter n. When the option is set, the software will prompt for a choice. In *sangnaka* letter representations such ng, mb, nd, ng as this option will prompt.

Eg:

kanda : කන්ද 'nd' is not a *sangnaka*

kanda surindu : කඳ සුරිඳු Here, 'nd' is a *sangnaka*

An expert user can set this option OFF for convenience. Then it is required to type *sangnaka* in capital and 'z' to represent 'ං' character.

Eg:

kaNDa suriNDu : කඳ සුරිඳු

Sizhala panthiya : සිංහල පන්තිය

- No Alt-key typing required.

A certain user may be familiar with *Wijeskara keyboard*, but it does not generate quality Sinhala fonts. These characters are stored at the end of the ASCII allocation table. Therefore if MSWord is used,

ව + ි will form වි

ම + ී will form මී

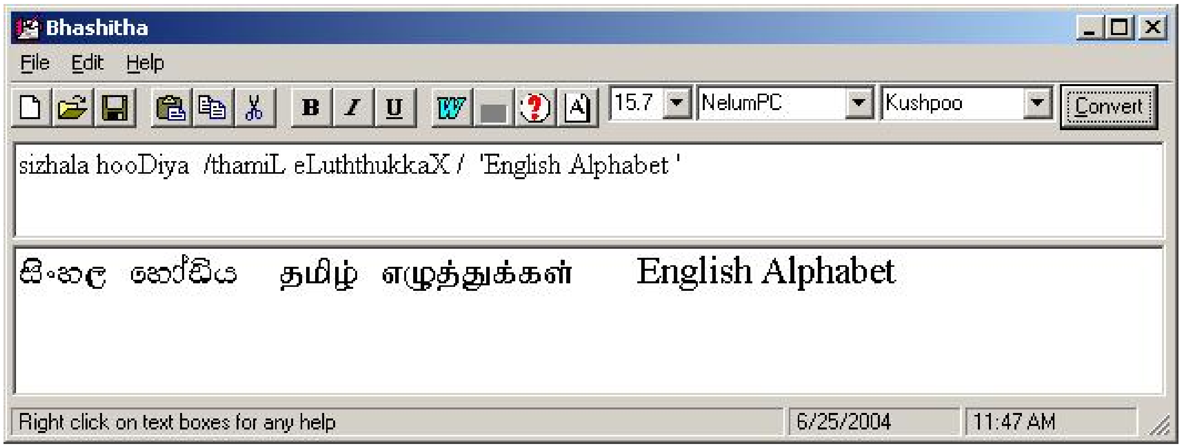
ර + ් will form ර්

Bhashitha would provide the best characters available in the font itself.

vi	: වි
mii	: මී
r	: ර්

- Simultaneous typing capability of Sinhala, Tamil and English

By default, the software converts the user input to Sinhala. The user can include Tamil and English letters in the same document without bothering to change the fonts. The only requirement is to type the Tamil content within slashes (/) and English content within single quotes ('). The software applies the necessary conversion methods and fonts.



sizhala hooDiya /thamiL eLuththukkaX / 'English Alphabet '

සිංහල හෝඩිය தமிழ் எழுத்துக்கள் English Alphabet

Right click on text boxes for any help | 6/25/2004 | 11:47 AM

- Direct link to MSWord

The user can export the working document content to MSWord for further word processing. Dynamic appending to the word document can be done by 'Send to MSWord' command. *Bhashitha* itself provides the facility to save and retrieve both the contents (pre conversion text and converted) in Text format.

- Punctuators and Numbers recognition

The user can type punctuators and numbers within the text itself without changing the font. But the conversion process will identify them and will provide the appropriate characters according to the font capabilities. If the font does not provide those characters, user can include them within single quotes (‘).

Eg: 2002.08.10 dina pe.va. 8.30 ta gadya'/'padya rachanaa tharaNGa pEvEthvee.
2002.08.10 දින පෙ.ව. 8.30 ට ගද්‍ය/පද්‍ය රචනා තරඟ පැවැත්වේ.

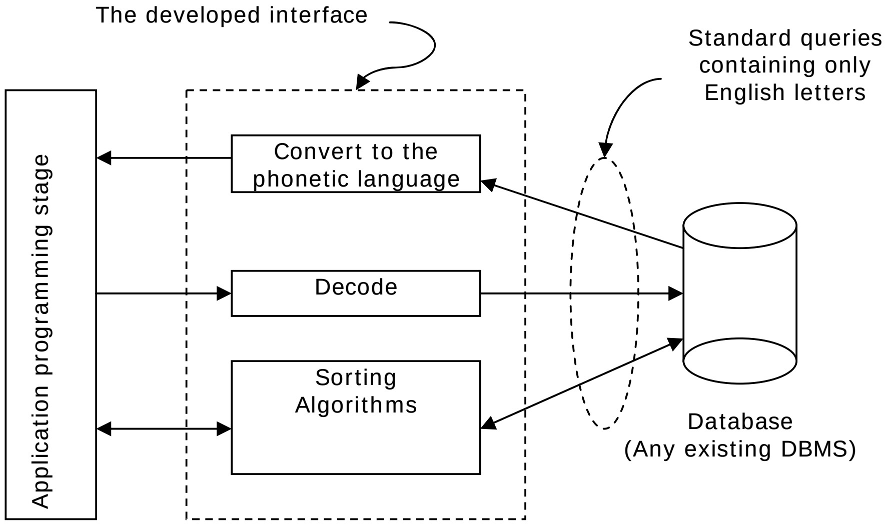
3.11. DYNAMIC GUI CONCEPT

This is the concept developed in order to manage the multi language user interfaces in software applications. This enables the programmer to handle GUI for different languages in a generalized manner and removes the requirement to design different forms (in programming terms) for each language. The concept can be explained as following.

- Design the layout of the form(GUI) and name visual components with array notification
- Keep a global indicator for the language
- All the captions and screen information for different languages should be stored as phonetic codes in a database or hard code.
- Make an interface which can access the stored information and conversion methods described above.
- Let that method to convert the provided information with relevant GUI parameters (font and appearance settings) and make the GUI dynamically.

3.12. DATABASE APPLICATIONS – ARCHITECTURE

The aforementioned conversion, decoding concepts and sorting algorithms can be directly integrated with the existing software tools and verity of database applications could be made with different language interfaces. (Sinhala,Tamil etc.) A generalized architecture can be explained for database programs,


The developed interface
Standard queries containing only English letters
Application programming stage
Convert to the phonetic language
Decode
Sorting Algorithms
Database (Any existing DBMS)

Generalized Architecture for database applications

3.13. E-MAIL APPLICATIONS

The phonetic concept and conversion methodologies so far explained provides an effective and economical solution to Sinhala/Tamil e-mail applications that are transmitted through IP network. It has been observed that major incompatibility of existing Sinhala font is their inability to make proper communication with browsers and web applications due to the spread through out the ASCII set.

The user input of Sinhala/Tamil message can be coded to the characters described in section 3.1,3.2 and that unique code can be used to communicate in the web interface. Therefore the e-mail server requires no change as it communicate with standard letters. The interface designer can capture a keyword in the message content and he can make the user interface to appear in required language using relevant conversions.

3.14. EXTENSION TO SMS

SMS(Short Messaging Service) and MMS(Multimedia Messaging Service) applications can be made based on the phonetic conversion principle. Two method are described here; one has been already implemented.

- Method 1 : a server application

In this method, the sender should send the desired message to the service provider with the destination number. Then a server application will analyze the message, do necessary conversion and make an image with the required language. Then the image will be sent to the receiver as an MMS or SMS. If SMS is used, the number of letters is badly reduced due to the pixel limit of the SMS picture messages.

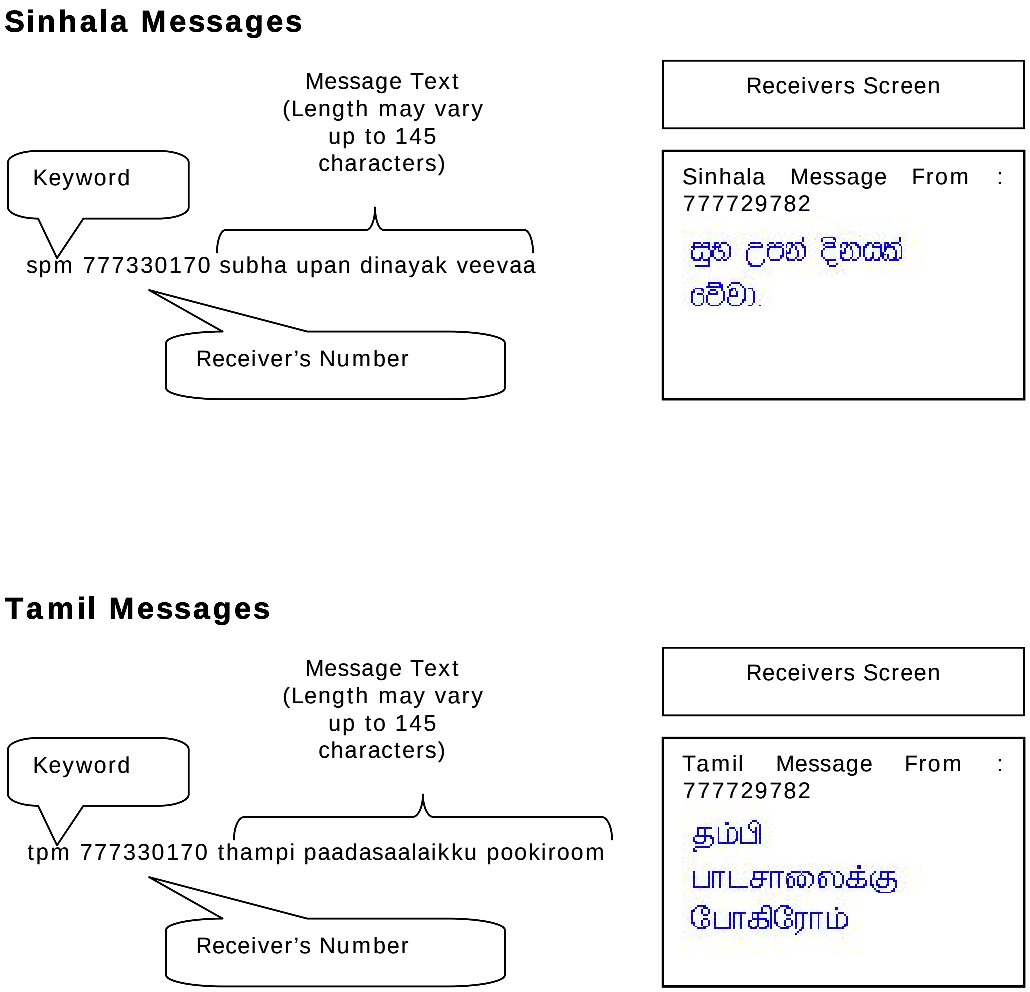

Sinhala Messages
Message Text (Length may vary up to 145 characters)
Keyword
spm 777330170 subha upan dinayak veevaa
Receiver's Number
Receivers Screen
Sinhala Message From : 777729782
සුබ උපන් දිනයක් වේවා.
Tamil Messages
Message Text (Length may vary up to 145 characters)
Keyword
tpm 777330170 thampi paadasaalaikku pookiroom
Receiver's Number
Receivers Screen
Tamil Message From : 777729782
தம்பி பாடசாலைக்கு போகிரோம்

Table 3.14-1 SMS/MMS Implemented Application - Example

- Method 2 : a phone application

In this method, the user will be provided with an interface to type the message in an interface where the converted preview will be available for the sender too. Therefore it requires a phone application which can handle the conversion process. The message can be sent as phonetic characters or as the generated image. If phonetic characters are sent, the receiver should also have a phone application to view the message in the desired language. Since most of the mobile phones are not capable to install Sinhala fonts, it is required to include a bitmap drawing mechanism to each letter in the language. The map can be stored in a text file in the phone and pixels can be drawn sequentially to generate letters.

3.15. TEXT TO SPEECH

A text to speech application can be made using the phonetic concept since there's a direct relationship with writing and pronunciation in phonetic languages. The input may be in a Sinhala font or according to a convention as mentioned in table 3.2.1. The speech generation can be done as dynamic combination of vowel and consonant sounds using waveform processing or a direct map to sounds of each combined letter. The sounds of combined letters can be stored in a single sound file and a separate map can be used to track the positions of each sound. If the Sinhala/Tamil string is coded to that map as the same procedure in the sorting algorithm, an effective method of text to speech can be achieved.

3.16. SPEECH RECOGNITION

As described in section 3.15 the relationship of the pronunciation and writing pattern of a phonetic language can also be used to recognize the speech. This kind of system requires wave form matching mechanism which has already available in the software tools. The generated code will not be as exact to the phonetic code, but an efficient map can be created by observing the patterns. When a map is created as the generated phonetic codes (Table 3.2.1) can be used to store the content and then they may be used for any other purposes.

3.17. OTHER RELEVANT APPLICATIONS

The above described methods and tools can be used as the base for number of designs and applications in the field of IT. As an example dictionary system can be developed to for Multi-linguistic functionality by storing information on the phonetic codes and manipulating them with previously described conversions. This concept will be a tool to develop a language translation concept among Sinhala-Tamil-English (or any other phonetic language) since it provides information manipulation without effecting to software model development.

4. CLAIMS

(1) A phonetic language computer interface system,

Which analyzes the phonetic language inputs according to their formation of vowel-consonant combinations and,

Handle the language (phonetic) information using a conventional character set assigned based on their vowel-consonant combination where,

Employs sorting algorithms based on the phonetic formation of letters by assigning weights to individual letters in order to ,

Provide a user friendly and programmer effective interface for language handling without bothering the writing conventions of the language.

(2) An interface as claimed in claim 1 where the phonetic language is Sinhala.

(3) An interface as claimed in claim 1 where the phonetic language is Tamil.