A Dynamic Text Input scheme for phonetic scripts like Devanagari

Amit Rathod
Interface design group
Media Lab Asia, KReSIT building
IIT Bombay
powai, Mumbai – 400076, India
91-022-5722545-exe5889
arathod@media.mit.edu

Prof. Anirudha Joshi
Industrial design centre,
IIT Bombay
powai, Mumbai – 400076, India
91-022-5722545-exe7829
anirudha@idc.iitb.ac.in

ABSTRACT

Devanagari text input presents unique challenges to the field of human computer interaction. We introduce a new scheme for phonetic scripts such as Indic, using a dynamic text entry approach for the keyboard which can also extend to a soft form. In this proposed scheme for text entry for the *Devanagari* script, all the *matras* graphemes are placed in the honeycomb of the surrounding keys and become dynamically available when any consonant character key is accessed. This scheme allows rapid input because of reduced finger travel. The aim is that everyone who can read should be able to type. This new *Devanagari* keyboard is targeted at the novice users and can be learnt quickly, retained and fair speed can be achieved. This design which give clear justice to the script structure inherent to the *Devanagari* which is radically different than the roman script. Hence need a paradigm shift in the way of text inputting. We have developed a prototype for *Devanagari* which has only the consonant keys and thus one need to hunt from a lesser keys leading to quick retention and reduced hunting load. Preliminary tests indicate that it is very easy to use with little training.

KEYWORDS

Dynamic keyboard, *Devanagari* text-entry, self-disclosing *matras*, Indic scripts, and Regional language interfaces.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires specific permission and/or a fee."development by design" (dyd02), Bangalore © Copyright 2002,ThinkCycle

1. INTRODUCTION

Text entry is one of the most frequent human computer interaction tasks. Although great strides have been made towards speech and handwriting recognition, typewriting remains and will likely be the main text entry method in the future. Computing devices demand text input schemes that can be quickly learnt and retained to achieve a fair speed and easy to use rather than "hunt and peck". Roman Keyboards are not particularly amenable to accommodate the phonetic non-alphabetic script like Indic. Many alternate layouts exist which are mapped on the Roman alphabets of QWERTY keyboards. However, no such known work exists for Indic scripts. Localization to Indic scripts is a non-trivial task due to significant differences from Latin based scripts and writing systems. Here we propose a new keyboard.

1.1 DEVANAGARI TEXT INPUT METHOD

In this section we investigate existing input method and what is good in them

The typing experience, in any language, involves a complex combination of mental and physical coordination. Beginners and experts alike seem to develop proficiency with the keyboard through an association of the phonetic value of a letter with a location on the keyboard. That is, when preparing to type a word, the word is analyzed and phonemes are broken down into letters. This is not always the way letters are processed; some may be referred by the physical shape of the letter rather than its sound value. Most typists, including many native speakers of Hindi, Marathi, are accustomed to the American Standard (QWERTY) keyboard layout. Those who must write in *Devanagari* often has to learn different

overlays on Standard QWERTY layout which are *Devanagari* character map like.
1. INSCRIPT (iLeap- C-DAC),
2. Aakruti phonetic
3. Typewriter layout (Godrej and Ramington)
4. ITR (Indian Typographic research)
5. Desha (NCST)
6. Transliteration (e-patra)

Transliteration: Its representation of *devanagari aksharas* with the closest phonetic equivalents of Roman script. It facilitates *Devanagari* typing using QWERTY keyboard. It follows phonetic approach which is so far most easy to handle for those who know English language on and well versed in typing on Roman keyboard. Most of the e-patra portals, web duniya, rediffmail regional language email service uses this mode. *Devanagari* display character set is mapped as closely as possible. It isn't as easy as it seems, for the way one has to type in English is not exactly 'natural', in terms of phonetics.A lot of absurd rules govern the exact transliterations which no way can be related to novices' mind.

Nobody wish to spend months of learning for these layouts because of casual approach. However, the most widely propagated platforms only offer support for these layouts. Though this support is implemented quite well and at the operating system level, there is no easy way to type in *Devanagari* for those who are not familiar with the these layout.

1.2 HOW THEY FAIL TO SUIT NOVICE REQUIREMENT?

Most of these layouts are designed keeping in mind the finger dynamics of the touch typing on QWERTY and frequency count. But for a novice or occasional user who predominantly shows single finger (or one finger of both hand) typing habit these options leave in so much of hunting.
And our stress is on those second generation user who don't have a precedence of English typing because they never have to learn English. For them there is no cognitive reference to hunt a key increasing the reaction time to a frustrating level.
With this proposed scheme the user can also start afresh keeping the habits of the single finger typing and still can achieve a fair speed to serve his purpose.

2. INDIC SCRIPTS

2.1 Phonetic basis of Indian languages

Most of the 18 major languages spoken in India have their orthography derived from the ancient *Brahmi* script. The others are Perso-Arabic in origin. *Panini*'s phonetic classification of the Indian alphabets into vowels (V: A,, Aa…) and consonants (C: k K ga Ga…) serves as a common base for all Indian languages derived from *Brahmi* scripts. In addition, there are also a few graphical signs used for denoting nasal consonants, nasalization of vowels etc (G). This scheme is phonemic in nature. Figure 1 show the different alphabets of our encoding for the *Devanagari* script. The effective unit of the writing system for all these Indian languages is the orthographic syllable, consisting either of a lone vowel, optionally, followed by a graphical sign with the structure (V)(G:Aa ^^M) or a consonantal syllable consisting of a consonant and a vowel (CV:ku) core and, an optionally following sign
(G: ^^M^). The canonical structure for a syllable is thus of the form (CCV (G: skao), as listed in the Unicode Standard (or [C]* CV [G]* in standard regular expression format). Two consonants in a syllable is a common phenomenon. In some syllables, the number of consonants can go even up to five.

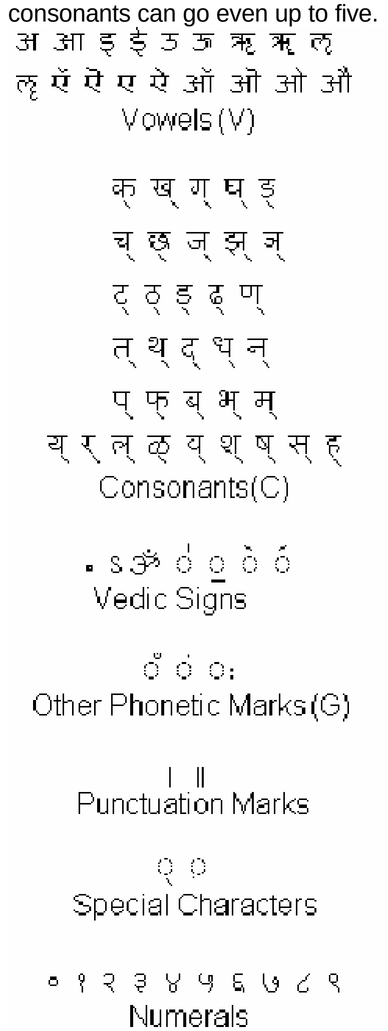


Fig2: Alphabets of *Devanagari* script

2.2 INDIAN LANGUAGE TEXT INPUT

There are three different approaches to Indian language text input for digital systems. These are Direct Entry, Graphic Entry and Phonetic composition. Direct entry is a direct adaptation of Hot Metal technology methods that were used in earlier days in printing Indian language texts. The basic idea is to have various type forms, in a font, corresponding to graphically distinct syllables which are then linearly composed to form words. Graphic entry is similar to the manual typewriter approach. Here a minimal set of graphic primitives is provided and syllables are composed as combinations of these primitives. Phonetic composition is a scientific approach to the composition of syllables. This method is endorsed by Unicode [12] and is rapidly becoming the input method of choice. Here consonants, vowels and other characters are encoded separately and syllables are dynamically composed at run time. Phonetic input has three variants – full consonant, pure consonant and transliterated Roman. The former two are distinct encoding techniques while the latter is just a convenience mapping. The variants *full consonant* and *pure consonant* arise because the same vowel may appear in different graphic forms depending on the context. Indian languages are vowel dominant. The consonants in their pure form do not include the vowel sounds. Vowels have to be attached to the consonants to make them complete. In the *pure consonant* approach, consonants are always in pure form and the vowels (including the 'A') are always explicitly added to the consonant to make it a full consonant. In the *full consonant* approach, the consonant is assumed to have the vowel 'A' attached to it by default. Grammatically, another vowel cannot be attached to it. One uses the *matra* forms of other vowels as a separate entity which attaches to the full consonant The full consonant approach needs encoding of a minimum of 66 basic primitives whereas the pure consonant approach needs only 50. A complete discussion of the two different encodings is provided in [8].

3. PROPOSED INPUT SCHEME

In this dynamic keyboard the first two rows accommodate all the consonants and semivowels, and thus require lesser number of keys to keep the continued focus on.

For any key pressed, the honeycomb of surrounding keys gives the vowel *matra* positions of which are consistent with the script model. Thus there is no hunting act for the *matras* and finger travel is reduced to half.

Fig3: the dispositions of the vowel *matras* around any consonant key pressed in unshift and shift level

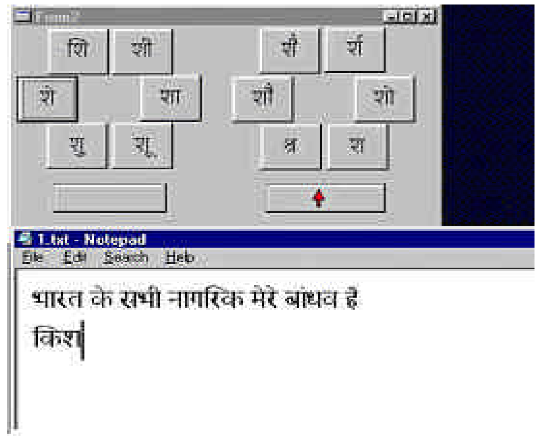
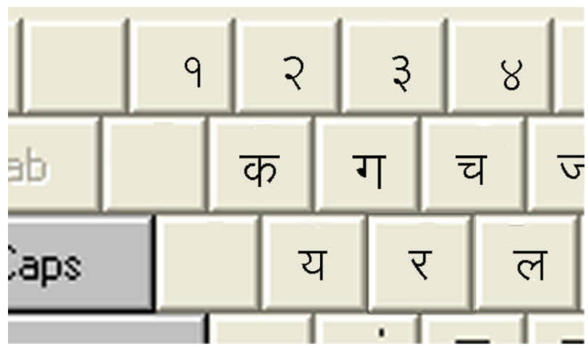


fig4: The two rows form the consonants and the semivowels and the vowel *matras* are picked I the runtime



Unshift

With shift

Fig 5: Key press turns the surrounding keys into vowel *matras* till the next key is pressed

3.1 CONTRIBUTIONS OF OUR SCHEME (Neighborhood floating vowel *matras* in dynamic keyboard)

Our dynamic keyboard has the following significant advantages over conventional keyboards.

1. Our keyboard requires lesser number of keys to keep the continued focus on. Our keyboard has 2 rows for all the consonant keys. It allows better focus and easier operation.
2. Consonants are grouped together i.e.phonetic *vargas*, as per their source points in the oral cavity. It eases the learning curve and facilitates faster hunt and peck once got the concept of vowel *matras* imbibed.
3. You cannot err / one way foolproof cannot type what you cannot pronounce in terms of multiple floating vowel *matras*. This feature can work well with phoneme feedback
4. Phonetic feedback can also assist for the blinds or speed typist who wants to read simultaneously.
5. Key location memory load is reduced rather now user can tell where a vowel *matra* key would lie without fail. Picking any vowel *matra* is almost "no thinking-hunting act" and makes both the hands free to pick consonant alone.
6. "Digraph frequency" shows that it is less likely to have consonant from the same *vargas* without vowel *matras* one after the other. In case of no *matra* to the earlier key, the *aksharas* in the neighborhood can be typed with typing the same key twice. The whole objective is to minimize the finger travel.

4. EVALUATION

Research proposition: **Attributes that matters for virgin user or secondary user:**

- hunting load should decrease
- learning over a period of time
- reduced finger travel and fair typing speed

Parameters that forms the basis of evaluation:

- Error rate (missed characters / in the particular paragraph)
- No of times backspace key is used
- Response time (average)
- Speed of typing (character rate)
- Ultimate speed one can reach
- Time taken to type a particular paragraph
- No of keystrokes
- Effort value E=f.x (applicable to touch typing)
- Finger travel (for individual finger)
- Iterations of look between the monitor and keyboard
- easy to learn (no of sessions taken to reach a predefined speed)
- easy for the novice to use : fast learn (achieve a fair speed after some 4-5 sessions of 20 mins)

The parameter that needs to be compared as an index of performance can be derived out of the research preposition. Hence we narrow down to the following theoretical evaluation indices

- Speed of typing (syllable rate)
- Time taken for text input
- No of times the backspace is used (error rate)
- Speed reached after some 5 tests for exposure spaced in time of 4 days

4.1 EVALUATION METHODOLOGY

We have done systematic evaluation in which we followed the following protocol and the approach. Initial user feedback in the pilot testing, however, has been very encouraging. Users familiar with the script have been able to input text with just a simple briefing. We feel that our scheme will require very little training. A few hours of practice should suffice to attain proficiency in this scheme. Once the conceptual model is formed about the vowel *matra* positions, the job of typing becomes fast and easy.

Usability study and learnability tests were performed. For this the subjects for usability testing (users) were secondary school Hindi teachers who have got sufficient exposure to the script and knowledge of the language. Functional Prototype is developed .We have implemented our current scheme for the Windows98 desktop for the '*Dynakey'* for dynamic keyboard. Our system is fully compatible with Unicode character encoding. One can perform all the Text editor commands.

4.2 USABILITY APPROACH

Prior to a testing with the subject, an "expert" user (here in this case the designer who is familiar with the new design) is recorded performing a task. The recording becomes a performance baseline. Later during actual usability testing, a "novice" user is recorded performing the same task. The action recordings of the two users are then compared and results are shown graphically. The hypothesis is that by graphically comparing the actions of an expert and average novice users, a usability analyst can quickly figure out where usability problems (e.g. confusion with key choices) arise with the text input device.

It is well known that there are many benefits to evaluating a user interface for usability. One method of evaluating a user interface is through usability testing. The testing involves observing a typical user performing predefined tasks with a system. Various types of information may be recorded including the time it takes to perform the task, the number and types of errors made, and the user's rating of the system. Video recordings of the user sessions are made. This data is then analyzed to identify problem areas in the keyboard layout. This analysis is largely a manual process and can be quite time consuming.

We record the developer performing the usability tasks and use their performance as a baseline for defining the desired performance of a novice user. Since the developers of the system typically know the optimum way to perform a task, a large deviation from this optimum performance may signify a problem area to look into it radically. We are still exploring how to graphically represent the time spent between actions, with the knowledge that a large pause between actions may represent time the user spent trying to decide what action to try to find next key or strategy to type some grapheme and therefore may indicate a usability problem for text inputing.

Also the existing keyboards in used by large no. of DTP operators and others is located which happened to be Inscript next to phonetic (transliteration). Hence performance as compared to Inscript keyboard becomes an additional baseline for benchmarking.

Results of the evaluation study will be formally analysed and pilot testing shows some promise as it comes close to the performance by the inscript. But in later tests there is fair amount of retention and learning is observed.

5. CONCLUSION AND FUTURE WORK

We have proposed a new input scheme that exploits the phonetic grouping and orthographic *matra* model inherent to script. It is smart, easy to use and logical. This scheme seems to reduce finger travel and cognitive load in hunting. We also think that the scheme may also be useful as an in virtual keyboard for conventional desktop systems and mobile stylus driven applications. Future plans are to extend this concept through

- extensions to other Indic scripts and IPA(International Phonetic Alphabets)
- Deployment and testing on actual situations.
- Use of the LCD display keys to assist the first time user.

6. REFERENCES.

[1] IS 13194, Indian Script Code for Information Interchange ISCII. Bureau of Indian Standards, Dec 1991

[2] Shyam Narayan, Prof Anirudha Joshi (IDC report) -Indian language computing, Year 1999-2001

[3] Bapurao Naik: Typography in Indian language Volume 1and 2

[4] Shrinath Shanbhag, Prof. Durgesh Rao, Prof. R. K. Joshi -NCST- An Intelligent Multi-layered Input Scheme for Phonetic Scripts, Symposium on Smart Graphics, June 11-13, 2002, Hawthorne, NY, USA

7. CREDITS

1. The project was initially a part of IDC Project 3, IIT Bombay by Mr. Amit Rathod under the kind guidance of Prof Anirudha Joshi and Prof. Uday Athavankar and now is being continued as an activity in Media Lab Asia, IIT Bombay hub-interface design group.
2. Initial usability tests are results of work of two Media lab asia interns

   Shruti Srhivastava, IV year B.Des.student, IIT Guwahati

   Gopal Vaswani, IV year B.Des.student, IIT Guwahati
3. The implementation and debugging was done by Dr Hayatnagarkar , Ph.D. IIT Bombay,Abhinav Gupta, IIIrd year B.Tech.student, IIT Kanpur