# A MULTIMODAL INDIAN LANGUAGE INTERFACE TO THE COMPUTER

*Hema A Murthy, C Chandra Sekhar*

Dept. of Computer Science&Engineering
IIT Madras, Chennai - 600 036

*C.S.Ramalingam, Srinivas Chakravarthy*

Dept. of Electrical Engineering
IIT Madras, Chennai. 600 036

**ABSTRACT**

Although India's average literacy level is about 65%, less than 5% of India's population can use English for communication. And even though the world-wide web and computer communication has given us access to information at the click of a mouse, 95% of our population is excluded from this revolution due to dominance of English. To overcome this problem we propose to set up an Indian Language Systems Laboratory at IIT Madras. Our initial goal will be to develop a *multimodal* interface to the computer that is relevant for India, i.e., one that enables Indic computing. The components of this Indian language interface will be:

1. Keyboard and display interface
2. Speech interface
3. Handwriting interface

## 1. INTRODUCTION

Imagine a villager walking into a rural Internet kiosk, who may be semi-literate or even illiterate, wanting to use the power of the Internet to either communicate with a relative somewhere else, or contact a city hospital, or get vital crop information. The currently available English-based keyboard and applications are totally unfamiliar and intimidating. As a result he or she feels shut out and not part of the ongoing information revolution. On the other hand, had the computer been able to accept speech input, and has applications in the local language, this typical villager would have been as comfortable using it as anyone else.

Computers have become an essential part of many facets of our lives. However, in the Indian context the use of computers is far less compared with that in the developed nations of the West because of the reason we have already hinted at: the language of the interface is almost always English and the communication is in the "written" form, i.e., via the keyboard. Barely $65\%$ of our population is literate, of which only an elite minority ($\approx 5\%$) can read, write, and speak the English language. This shuts out most of the Indian population from the world wide web and its huge potential. Therefore it is essential to have an interface that uses not only the local language but also speech, to cater to the needs of the semi-literate and illiterate sections of the population. Moreover, the current keyboard has been developed for English and cannot be naturally adapted for Indian languages. Hence there is a need for a handwriting interface too. We call such an interface as being *multimodal*. In the West, although speech and handwriting interfaces are available for English, these are for some specific applications, namely, dictation machines (hands-free), limited handwriting/graffiti recognition as in PDAs. This is primarily because of the simplicity of the Roman script. Whereas, in the Indian context, these interfaces must be part of main stream applications, namely mail readers, web browsers, word processors. This is the FIRST such effort to seamlessly integrate all three interfaces.

Developing the multimodal interface is a significant multidisciplinary effort. At IIT Madras (IITM) different groups have been working on aspects of the individual interfaces. To develop an effective solution it is essential for these groups to work cohesively toward a long-term goal over several years. Clearly, there will also be specific deliverables each year. We have set up an Indian Language Systems Laboratory at IITM with a view to developing a multimodal interface for Indic computing, drawing upon the already available expertise and collaborating with leaders in areas where we lack the needed capability. This multimodal interface has the potential to revolutionize Indic computing. Hence our approach and design will be shaped by the need to deploy it commercially on a wide scale. The larger goal of the proposed lab is to focus on various issues related to Indic computing.

The TeNet group at IITM has pioneered the increase in connectivity to villages by developing the corDECT wireless local loop technology []. Already more than one million connections have been established by n-Logue communications in the Internet kiosks it has set up throughout the country. Hence the need for a multimodal interface for Indic computing is not a theoretical assessment but one resulting from practical need. Moreover, it provides the framework for widespread deployment and use.

In the following Sections we will give a brief overview of the R&D work that is needed, identify the issues, and

highlight the expertise available at IITM to tackle them. We have also identified groups outside IITM with whom we wish to collaborate. The specific deliverables are given in Section 2.7.

## 2. THE MULTIMODAL INTERFACE

The multimodal interface interacts with the user via several input mechanisms (such as keyboard, mouse, speech, and handwriting) and several output mechanisms (such as display and voice). Ideally, the computer should automatically choose the most appropriate one, based on the context and the user. This very challenging task is our long-term goal. Initially, to simplify the problem, we will require the user to specify the mechanism, e.g., choose speech for the output.

Even here, error-free recognition of continuous speech independent of the speaker, speaking style, microphone, and surrounding conditions still lies in the realm of Science Fiction. What is practically possible with the current technology, given the conditions in a typical kiosk, is achieving good accuracy on limited vocabulary tasks. Similarly, the quality of current synthesizers is such that they have only limited commercial acceptability. However, there is ample scope for improving the performance of these interfaces, not only by just improving the basic technology, but also by cleverly designing the man-machine interface (MMI) such that their effective performance increases, leading to wider user acceptability.

### 2.1. Input Mechanisms

#### *2.1.1. The Keyboard and Character Encoding*

Keyboards that are easy to use for Indian languages are possible only if character encoding is based on a well-thought-out design. The design issues become even more critical if we have to deal with multiple languages.

Today, applications in local languages are beginning to appear. But they are not only language specific but also dependent on the character encoding and font used! We have fifteen official languages and many more unofficial ones. With the existing approach, developing language-specific applications is a Herculean task. Instead, the correct approach will be to develop applications *independent of the language, font, and encoding by making changes at the operating-system level*. If this is done, all applications will inherit the new interface, obviating the need to recompile whenever the language is changed.

We have already begun work on such an interface in which modifications have been made at the kernel level, eliminating the need to recompile an application whenever there is a language change. Changing the language is simply a matter of changing configuration files. All input passes through a language filter. Further improvements will be done as part of the proposed work.

Non-uniform encoding is another important issue for Indian languages. In English, one keystroke corresponds to one letter and there are no character clusters. But in a typical Indian language there are roughly 3000 character clusters. Therefore, the number of keys on the keyboard will become unmanageably large if we want a single keystroke for each cluster. Instead, we will have to make do with a sequence of keystrokes. As a result, the encoding will not be uniform for all character clusters—it can vary from a single byte to three bytes. Since the number of characters in the script corresponding to every language is different, we propose to support different encoding schemes—for example, mapping could be made dependent on the frequency of usage of a cluster in written text.

Finally, the present keyboard is designed for English. The efforts for developing keyboards for Indian languages has been along the lines of remapping the keys of the QWERTY keyboard. This is a very unnatural adaptation. Instead, we propose the keyboard be designed from scratch, keeping in mind the similarities that exist between all the Indian languages. If there are important differences, those must also be taken into account and localized accordingly. For example, Tamil does not have aspirated consonants, which frees up keys that can be used for some of the most frequently occurring character clusters.

#### *2.1.2. Speech Recognition*

The goal of this input interface is to accept speech and take appropriate action by recognizing what is spoken. The action could be anything that the current keyboard interface is capable of, e.g., change directory, open a file, etc. However, certain kinds of tasks are more easily carried out using the keyboard. Hence the speech interface will coexist with the keyboard and not replace it. Clearly continuous speech recognition is the core technology for the input interface.

To be able to make progress in speech recognition having standardized databases is an essential pre-requisite. One of the major issues in Speech Recognition for Indian languages is the lack of such databases. In the US, a variety of databases for English have been collected over two decades, in a variety of conditions (clean speech, telephone, cellular, etc.), spanning different applications. By contrast, we do not have a collection that is even remotely close for even one Indian language. Collecting such databases is a significant effort and we need a concerted effort to rectify this basic deficiency. We would like to pioneer the Indian language speech database collection. Initially we would like to collect data for two Indian Languages, viz. Hindi and Tamil, and then work toward data collection for all Indian Languages. Databases are also required for speech synthesis, but the requirements are different (see also Section 2.3).

Before collecting a database, it has to be carefully designed by an expert linguist, taking the application into consideration. Annotation of the collected data requires careful development of a number of tools (see Section 2.4).

### *2.1.3. Handwriting Recognition*

Although Indian language computing will significantly increase number of non-English speaking public to use the computer, the corresponding keyboard is very cumbersome to use. As we pointed out earlier, a typical Indian language has roughly 3000 character clusters (here by character cluster we mean the $C^*V$ or $V$ ($C$ stands for "consonant" and $V$ for "vowel"). On a keyboard that is restricted to 256 possible scancodes, it is not possible to avoid multiple keystrokes for most of the clusters except perhaps the most frequently occurring ones. An attractive alternative is the handwriting interface.

Handwritten character recognition is of two varieties depending on the means used to produce it: (i) scanned image of handwritten text on paper, and (ii) handwritten text produced by an electronic pen, where the pen trajectory on a special tablet is processed by the computer. The latter is known as Online Handwritten Character Recognition (OHCR). A handwritten character is composed of a set of pen strokes. A stroke is a line drawn by a pen between the time when a pen touches the writing surface and the time when it is lifted. Therefore, identifying the component strokes of a character is the first step to character recognition. The following issues need to be addressed:

- Database of Strokes: Stroke databases for all Indian scripts is not available. We would like to collect the needed databases. We would like to exploit the structural similarities among various Indian languages by using stroke databases shared among multiple languages and come up with a smaller-sized solution.

- Output format: Currently ISCII is a widely accepted standard for representing Indian language characters. More recently different standards have started emerging for different Indian languages. In this context, we will borrow from the effort on keyboard and display interfaces to produce output based on the encoding used.

- Interfaces: An OHCR system consists of four main components. The 1) input area where the writer enters content, 2) the stream of stroke IDs, 3) the stream of characters, and finally 4) the output. Between every stage and its successive one, interfaces must be defined and standardized. Standard interfaces are the key to the longevity of a software.

## 2.2. Output Mechanisms

In this section we discuss different output mechanisms. Although humans have five different sensors the focus here is on the visual and auditory sensors.

### *2.2.1. Display*

Unlike in English, the display corresponding to a keystroke is not only dependent upon the past keystrokes but also on the future. Enabling such support requires a language model for the representation of graphemes.[1] For this, the kernel must be enhanced to support such language models. Additionally, the widths of the graphemes vary significantly, making variable-width fonts essential (fixed-width fonts will give ugly results). There are a number of issues associated with variable-width fonts that we propose to address. For example, the width of the grapheme that represents $\eta$ in Tamil is four times as large of the grapheme that represents $r$ ($\eta$ and $r$ are symbols taken from the International Phonetic Alphabet).

As mentioned earlier, a sequence of keystrokes may have to be pressed to generate a single character cluster. Further, as one types, the cluster will have to be modified to ensure that it corresponds to the key sequence in progress. This brings in the related issue of the positioning of the cursor. The X-library that is currently available supports glyphs. Glyphs are bitmaps that correspond to the representation of characters in an Indian language. Multiple keystrokes can correspond to a single glyph and vice-versa. The X-Library currently supports a glyph based approach for cursor positioning. This is quite awkward when the 'delete' key is pressed. This results in a part of the cluster being deleted. Ideally one would like the code corresponding to that of the keystroke being deleted. Further, it will not work if the font is changed.

Currently the cursor positioning is handled by the application. Since most applications are developed for English/European languages, applications will have to be modified to support Indian languages. This would result in every application being modified. Alternatively, output methods can be added to the X-library which can be used by the application at the time of display, thus ensuring that the application is made language independent.

Once the interface is available, applications need to be customized for each language. Sometimes applications need to be modified to separate the language dependent parts from the language independent parts. In the current system, we have successfully completed the customization of KOffice in Tamil, Hindi, Marathi, and Malayalam.

[1] Graphemes are the set of units of a writing system (as letters and letter combinations) that represent a phoneme.

#### 2.2.2. Speech Synthesis

The speech output interface uses speech synthesis as its core technology. Similar to its input interface counterpart, it will not replace the monitor but supplement it. This is because there are many instances where the output is not text; even for text output it is more convenient in many cases to see the display on the screen rather than having it converted to speech and played out (especially when the size of the text message exceeds a few words).

Our goal is to synthesize natural sounding speech in all the official Indian languages. We will start with Hindi and Tamil.

Similar to the speech recognition effort in India although synthesis of intelligible speech is a simpler task than recognition, the major lacuna seems to be the nonavailability of databases and a systematic linguistic analysis of languages. See Section 2.3 for mored details.The databases required for synthesis are quite different from those required for recognition. Nonsense words spoken by a single speaker with minimal variations in pitch is typical.[2]

Significant linguistic expertise is needed for language analysis and design of databases. Some of the issues that need to be addressed are:

- Choice of basic unit: How large should the basic unit be to enable unrestricted speech synthesis? Can the unit be a common choice for all Indian languages?
- Development of natural sounding speech: The naturalness of the synthesized speech crucially depends upon the prosody that is associated with it. What kinds of prosodic models have to be developed to produce natural-sounding speech?

Although a large number of studies are available from language departments across the country, there are no studies that are driven from the requirements of synthesizers. To this end, we plan to collaborate with The International School of Dravidian Linguistics (ISDL), Trivandrum, and Central Institute of Indian Languages (CIIL), Mysore, and other Institutes, wherever such effort is available, to develop prosodic models for synthesis.

### 2.3. Developing Local Language Databases

As we pointed out in Section 2.1.2 one cannot overstate the importance of comprehensive, high-quality databases for making headway in speech recognition, synthesis, and handwriting databases. Since the multimodal interface is essential for the success of Indic computing, we strongly feel that a top national priority in this area is to create databases in all the major Indian languages. However, comprehensive database collection is a massive effort that cannot shouldered by any single group. This is because there is no universal database that can be defined since the requirement is problem-dependent. Hence the effort involves expert linguists designing databases based on the type of application, defining the different surrounding conditions under which the data are to be collected, setting up the collection points and collecting the material, and finally annotating the databases. This last part is perhaps one of the most time-consuming and laborious part. Some of this work can be reduced by developing annotation tools, which can semi-automate the process. All the above have to be repeated for each language of interest.

We would like to take a lead role in coordinating this effort across various groups around the country in establishing this national database.

[2]Whereas, for speech recognition we want the database to contain as many speakers as possible, speaking naturally.

### 2.4. Development of Tools

Once information is available in multiple languages, appropriate linguistic tools need to be developed for each language. We need dictionaries, spell-checkers, and language models, font conversion tools, encoding conversion tools, grammar models for parsing sentences, tools for signal processing, tools for image processing, tools for collecting speech data and annotating the same, speech recognizers and synthesizers.

### 2.5. Expertise at IITM

In this Section, we focus on the expertise available at IIT Madras to embark upon the task of building multimodal interfaces to the computer.

#### 2.5.1. Keyboard and Display Interfaces

The TeNeT group at IITM (www.tenet.res.in) started work on the Indlinux project about two years ago. We have made significant progress in developing Linux-based Indian language interfaces. We have modified the kernel and X-Libraries to support Indian languages.

Several multilingual applications have been developed by Systems Development Laboratory, Dept. of Computer Science and Engineering, IIT Madras (website: acharya.iitm.ac.in). A new fixed width code has been used to represent all the aksharas of Indian languages.

#### 2.5.2. Speech Interfaces

The Speech laboratory at IIT Madras has been working on Speech systems for the last 20 years (website:speech.cs.iitm.ernet.in). Similar effort has been started by the TeNeT group (website:www.tenet.res.in). This has resulted in the development

of a number of different tools, and some laboratory based systems.

#### *2.5.3. Handwriting Interfaces*

We have already developed an OHCR system for Tamil. Some work is done on online recognition of Telugu also. The experience gained in working with Tamil and Telugu scripts will be helpful in extending the work for other Indian languages.

### 2.6. Expertise Elsewhere

In this Section, we briefly describe some of the efforts in progress elsewhere, which can be used in the development of the multimodal interface. The groups mentioned below are keen in collaborating with the proposed lab.

#### *2.6.1. IIIT Hyderabad (*http://www.iiit.net/ltrc*)*

A team of engineers have been actively involved in machine translation at IIIT Hyderabad. The effort dates back to more than 20 years. They will be working with us on issues related to the development linguistic tools, namely, dictionaries, spell-checkers, grammar checkers.

#### *2.6.2. Chennai Kavigal(*http://www.chennaikavigal.com*)*

Chennai Kavigal is a Chennai-based company which has been developing local language software over the last six years. They now have a complete Office Suite in both Hindi and Tamil that works for the Windows environment. Chennai Kavigal will be involved in the productisation part of the multimodal interface. We are already actively collaborating with Chennai Kavigal on the Windows based effort.

#### *2.6.3. n-Logue Communications(*http://www.n-logue.co.in*)*

The success or failure of any product is governed by user-acceptability. To get the all-important user feedback, we propose to collaborate with n-Logue Communications, who is a Chennai-based rural ISP provider established about two years ago. Its mission is to set up Internet kiosks in towns and villages (excluding the top 150 cities). They represent the link to the end-users of the products of the Indian Language Laboratory. They will be be providing us the testbeds for the multimodal interface so that we get actual user-feedback, which essential for refining the design of the multimodal interface.

#### *2.6.4. Ministry of Information Technology Sponsored Efforts*

There are a number of sites (website: http://tdil.mit.gov.in) where related work is pursued. For example, at CDAC, the effort is primarily on developing keyboard interfaces. Further, different sites are focussing on a specific Indian language (RC-ILTS-Tamil, Anna University focuses on Tamil, while CI, IIT Bombay's focus is on Marathi). There is no single site, where the approach is to build a single **Multi-Modal Interface** to the computer. Wherever appropriate we would like to draw up these resources.

Given the track record of the TeNeT group (www.tenet.res.in), we do believe that the TeNeT group has the capability to transform such efforts into products. Further, we would like popularise these interfaces via n-Logue's kiosks.

### 2.7. Goals

Similar to the West, currently the Internet is most widely used for email and browsing. Therefore, our first goal will be to give the PC in a rural Internet kiosk the ability to use the multimodal interface for these tasks. This entails not only replacing the English keyboard but also adding speech and handwriting interfaces for ease of use by a typical villager. Initially we will concentrate on Hindi and Tamil. The design will be made as language-independent as possible. The PC will use the Linux operating system.

Our next goal will be to enable applications that use the multimodal interface, e.g., web-based applications, Office Suite. We propose to integrate the multimodal interface with *Chennai Kavigal*'s Office Suite product - Shakti 2.0. The next goal would be a more complete suite of applications, viz., voice-mail, video-mail, voice- and video-enabled e-mail, transliteration, limited translation and searching, and content-generating tools. Our ultimate goal is a PC. that automatically detects and adapts the characteristics of the user such as language, typing skills, literacy levels, and any other disadvantage the user may have (for example, the visually impaired).

## 3. CONCLUSION

It is hoped that development of a multimodal interfaces to the computer, will bridge the gap between the haves and the havenots and will go a long in the empowerment of the rural folk in India. With the proliferation of the Internet across the countryside, Mahatma Gandhi's dream of Village Swaraj can indeed become a *reality* (at least in the Information Technology sphere).