Web-enabled Speech Synthesizer for Tamil

P. Prathibha and A. G. Ramakrishnan
Department of Electrical Engineering, Indian Institute of Science,
Bangalore 560012, INDIA

---

1. Introduction

The Internet is popular and attractive because it gives a one-step access to practically any type of information. However, obtaining the information from the web requires sitting at the desk and reading, which is less desirable in a highly mobile society. Any time access to Internet content is sparking the demand for simple but personalized services that let individuals and businesses obtain fast and easy access to the people and the information they care about. Text-to-Speech (TTS) is increasingly being used to read out email and enterprise information to callers. Real-world applications in the local language, such as emails and read over the phone, are made possible with the help of Web-enabled speech synthesizer. In this paper, we report a web-enabled speech synthesizer for Tamil that reads HTML/text in Tamil language. Web-enabled speech synthesizer is made available as a plugin. The plugin consists of two major blocks: HTML parser and synthesis block. The HTML parser is responsible for gathering the text information from the HTML document specified from the user and delivering the text output to the synthesis block. Any text in TAB/ISCII/Unicode format appearing on HTML can be converted to speech. The synthesis block synthesizes speech. The plugin is designed to learn the format requirement of the user and then the synthesis system generates output in that audio format.

2. Development for Platform Independence

The Web-enabled speech synthesizer is coded in JAVA. The system is designed to read the text input through keyboard in TAB/ISCII/Unicode and to read the HTML/DOC files coded in TAB formats. The primary motivation for coding speech synthesis in JAVA [1] is the need for a platform-independent language that can be used in all machines independent of the operating system. The trouble with VC++ is that it is designed to be compiled for a specific target. However, with the advent of the Internet and the web, the old problem of portability has returned with vengeance. Even though many types of platforms are attached to the Internet, users would like them all to be able to run the same program. What was once an irritating but low-priority problem has become a high-profile necessity. So, while the desire for an architecture-neutral programming language provided the initial spark, the Internet ultimately led to Java's large-scale success. Accordingly, we have developed our speech synthesis application in JAVA.

3. The Technique

The web-enabled speech synthesizer downloads the HTML document, parses the HTML document to extract the text and feeds the text to the synthesizer and plays the synthesized speech. The block diagram of web-enabled speech synthesizer is shown in Fig. 1. The web-enabled synthesizer is in the form of a plug in. The input to the Plugin is the HTML document and the output is intelligible Tamil speech.

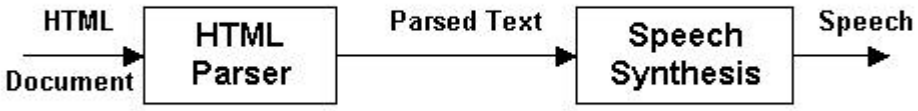

Figure 1. Basic block diagram of web-based speech synthesizer

The technique used in web-enabled speech synthesizer involves a two-stage process:

- HTML Parser

Speech Synthesis

The Plugin downloads the web page specified by the user and sends it to the HTML parser. The HTML parser retrieves the hypertext document coded in TAB/ISCII/Unicode format [8] and generates textual output in a form suitable for feeding to the synthesis block. The speech synthesis block converts the text, which is in TAB format into intelligible speech. Finally, the synthesized speech is played out on a speaker or a headphone in the format specified by the user. The user is also given the option of highlighting the text that he wants to listen and right click with the mouse. The option to select "speak" will be displayed and when the user clicks on it, the text is fed to the synthesis block and the speech is played out. Figure 2 displays one of the Tamil website coded in TAB format. Irrespective of the coding scheme, the web-enabled plugin is able to deliver web information as speech.

4. 1 The Synthesis scheme

A waveform concatenation [2] approach is adopted for synthesis, which is primarily used by Thirukkural [3]. In concatenation-based speech synthesis, natural speech units are concatenated to give the resulting speech output. These systems have more flexibility in selecting the speech segments to concatenate because the waveforms can be modified to allow for a better prosody match. It involves two phases, namely, the offline phase and the online phase. The offline processes of the system include: (1) Choosing the basic units, (2) Building the database, (3) Detailed study of prosody in natural speech, (4) Consonant-vowel segmentation, and (5) Pitch marking. These steps are explained below.

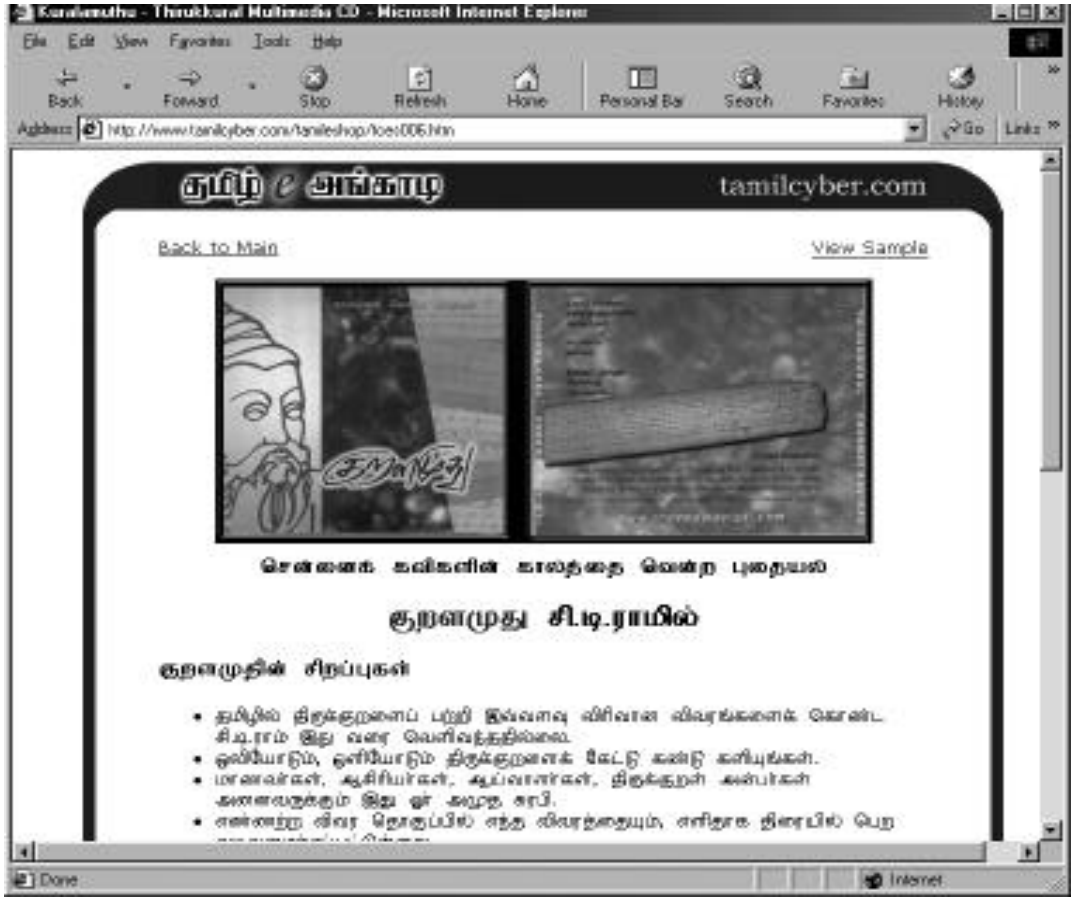

Figure 2. A sample Tamil website

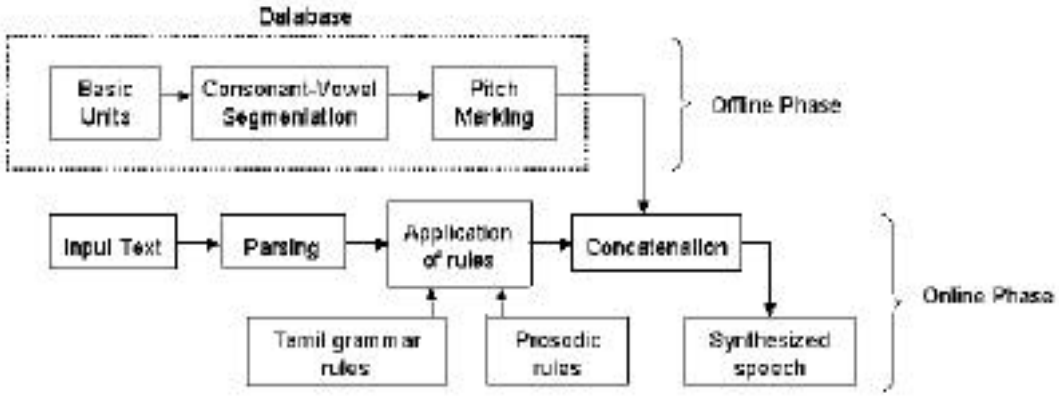

Figure 3. Block diagram of speech synthesis system using concatenation

Choosing the basic units

Basic unit of speech is a phoneme. Other units that can be used for synthesis are diphones, triphones, demi-syllables, syllables, words, phrases and sentences. In terms of quality, sentence is the best and phoneme is the worst. However, the size of the database also is an important factor to be considered. The issues in choosing the basic units for synthesis are:

- The units should lead to low concatenation distortion.
- The units should lead to low prosodic distortion.

- The units should be of a general nature, if unrestricted text-to-speech is required.

The different types of units that are used for our system are V, CV, VC, VCV, VCCV and VCCCV, where V stands for a vowel and C stands for a consonant.

Building the database

The database was collected from a native Tamil speaker over a span of several months. Recording took place in a noise free room using Shure SM 58 microphone, whose frequency response is 50-15,000 Hz**,** connected to a PIII – 500 MHz PC. Spoken units were recorded at a sampling rate of 8 KHz.

Detailed study of prosody in natural speech

Prosody is a complex weave of physical and phonetic effects that are being employed for expression. Prosody consists of systematic perception and recovery of the speaker's intentions based on pauses, pitch, duration and loudness. Pitch is the most expressive part of a speech signal. We try to express our emotion through pitch variation. Constant pitch signal sounds very unnatural. Therefore, a detailed analysis of duration and pitch is made and the information is stored as a look-up table for reference during synthesis.

Consonant- vowel segmentation

It is observed that any change in the consonant part of a signal results in change of perception of the unit. Consonants must be kept intact. To this end, consonant and the vowel regions of the units must be segmented.

Pitch marking

Pitch marking is essential as the waveforms are concatenated at the pitch marks. The method employed for estimation of pitch is DCT based spectral auto-correlation [7].

The online phase involves two important steps: (1) Text Analysis and (2) Synthesis.

Text analysis

Text analysis phase converts the input text into a sequence of basic units of speech and applies on them Tamil phonetic rules. It is also responsible for putting nonorthographic characters into representations that can be recognized by the synthesis system.

Synthesis

Specific language related phonetic rules are incorporated for synthesizing a natural sounding speech. Finally, Tamil speech is synthesized by concatenating the waveforms of the basic units in the correct sequence after the application of the prosodic rules [4]. Prosodic rules involve duration adjustment followed by amplitude modification. The waveforms are concatenated at the pitch marks. The pitch values may not match at the concatenation point. This is taken care of by applying pitch modification algorithm at the concatenation point [6]. At the point of concatenation, there may also be mismatch in the amplitude. The mismatch is

minimised by normalizing the amplitude to a certain fixed threshold. Thus the synthesis system generates intelligible Tamil speech.

3.2 HTML Parser

The function of the HTML parser [9] is to extract the relevant text and feed it to the synthesis block. The HTML parser consists of a HTML retrieval utility followed by tree based network and text parser. The java script is used to download the HTML document specified by the user. The HTML parser utilizes the tree-based network to separate start tags, end tags, text, images and comments from the HTML document. The text parser detects special text such as title and bold and italic types of font and type of coding such as TAB, ISCII or Unicode. The text parser also detects some of the redundant data like duplicate specification of email address in the HTML document and discards it. The text parser ignores images, video, frame information and postscript file. The text parser has the utility to expand the constituents of email addresses composed of alphanumeric characters. The text parser processes the email address separately by expanding the alphanumeric character and presents it in the form that can be used without ambiguity by the synthesis system.

3.2.1 HTML Parser Implementation

The HTML parser scans the source as a stream of start tags, end tags, text, comments, and so on. The parser extracts and builds a parse tree -- a tree-shaped network of objects representing the structured content of a HTML document. Once the document is parsed as a tree, the common tasks of extracting data from that HTML document/tree is quite straightforward. The parse tree [9] is represented as a network of objects -- the root, an element with tag name html. Its children typically include head and body elements, and so on. Each element in the tree is an object of the class HTML element.

So, if you take this source:

```
<html><head><title>Doc 1</title></head>
<body>
Stuff <hr> 2000-08-17
</body></html>
```

and feed it to HTML Tree Builder, it'll return a tree of objects that looks like this:

```
          html
         /    \
      head      body
     /        / | \
  title   "Stuff"  hr  "2000-08-17"
    |
 "Doc 1"
```

This is a pretty simple document. If it were any more complex, it would be a bit hard to draw in that style, since it sprawls left and right. The same tree can be represented a bit more easily sideways, with indenting:

- html
  - head
    - title
      - "Doc 1"
  - body

- "Stuff"
- hr
- "2000-08-17"

Both representations express the same structure. The root node is an object of the class HTML element, (Actually, the root is of the class HTML Tree Builder, but that's just a subclass of HTML element, plus a few extra methods like parse file.) with the tag name html, and with two children: HTML element objects, whose tag names are head and body. And each of those elements have children, and so on. Not all elements have children -- the hr element doesn't, for instance. And not all nodes in the tree are elements -- the text nodes ("Doc 1", "Stuff", and "2000-08-17") are just strings. That's all that there is to it -- one throws HTML source at Tree Builder, and it returns a tree of HTML element objects and some text strings.

However, what does one *do* with a tree of objects? We code information into HTML trees not for the fun of arranging elements, but to represent the structure of specific text and images -- some text is in this list element, some other text is in that heading, some images are in this table cell with those attributes, and so on. Thus we have separated the relevant text from the HTML document. The extracted text is then fed to the text parser. The Tree Builder approach is the most robust, because it involves dealing with HTML in its "native" format -- the tree structure that HTML code represents, without any consideration of how the source is coded and what tags are omitted.

4. Applications

Depending upon the content available, the potential applications of web-enabled speech are:

- Reading of e-zines.
- Access to check headline news in a local language.
- Access to retrieve and respond to daily emails, calendars, faxes and voice mails.
- Access to get local movie listings, critic’s review and to get driving directions to a customer site.
- Access to get up-to-the-moment sports scores, stock quotes and surf reports
- It can be used as an aid for the blind giving them access to information available on Internet.
- In most banking applications, speech technology can process and respond to routine inquiries from customers. For example, web-enabled TTS can recognise the name of the payee and process customer’s requests for a particular transaction. Since the name of the payee changes with each call, it is not practical to pre-record the answers to these requests. The combination of a wide range of answers to customer inquiries and the constantly changing data that makes up these answers requires TTS technology that can "read" dynamic information aloud.
- Self-learning multimedia education packages in Indian languages.
- Audio on-line help in all Indian language based IT software.
- The Java plug-in combination with word processor is a helpful aid to proof reading. Many users find it easier to detect grammatical and stylistic problems when listening than reading.

5. Conclusion

The web-enabled Text-to-speech synthesis system helps in retrieving the information through Internet by reading it out for the user. It can be easily downloaded and used for a variety of applications for synthesizing speech. It can also be combined with speech recognition models, which may be used as interactive voice response systems.

References

1. Herbert Schildt, *The Complete Reference – JAVA 2, Fourth Edition,* Tata McGraw-Hill Publishing Company Limited, 2001.
2. Douglas O'Shaughnessy (2000), *Speech Communication – Human and Machine, Second Edition*, IEEE Press, 2000.
3. G. L. Jayavardhana Rama, A. G. Ramakrishnan, V. Vijay Venkatesh, and R. Muralishankar, "Thirukkural: a text-to-speech synthesis system", *Proc. Tamil Internet 2001*, Kuala Lumpur, pp. 92-97, August 26-28, 2001.
4. R. Muralishankar and A. G. Ramakrishnan, "Human touch to the Tamil Synthesizer", Proc. Tamil Internet 2001, Kuala Lumpur, pp. 103-109, August 26-28, 2001.
5. Aniruddha Sen and K Samudravijaya, "Indian Accent text-to-speech system for web browsing", *Sadhana*, pp. 113-126, February 2002.
6. R. Murali Shankar, M. Anoop, T. Harish, A. K. Rohit Prasad and A. G. Ramakrishnan, "DCT based Pitch Modification", *Proc. Sixth Biennial Conference on Signal Processing and Communication SPCOM'01*, IISc, Bangalore, pp. 114 – 117. July 15-18, 2001.
7. R. Muralishankar and A G Ramakrishnan, "Robust Pitch detection using DCT based Spectral Autocorrelation", *Proc. Intern. Conf. on Multimedia Processing*, Chennai, pp. 129-132, 2000.
8. T N C Venkata Rangan and Satheesh, "XML and Tamil text processing with Unicode", *Proc. Tamilnet2001*, Kualalumpur, Malaysia, pp. 32 - 33, Aug. 26-28, 2001.
9. Ken MacFarlane's article "Parsing HTML*", A User's View of Object-Oriented Modules,* article number 17, Manning Publications.