Implementing a low-cost Speak-to-Dial service over VoIP

Ahmed Imran
TCS Innovation Lab - Mumbai
Yantra Park
Thane (West), Maharashtra
+91-22-67781612
ahmed.imran@tcs.com

Sunil Kopparapu
TCS Innovation Lab - Mumbai
Yantra Park
Thane (West), Maharashtra
+91-22-67788216
sunilkumar.kopparapu@tcs.com

ABSTRACT

Speak-to-dial (or Name Dialing) brings convenience to the phone users, who do not need to remember *people-phone number* associations to make a phone call. There are several name dialing solutions implemented and used in personal as well as work scenarios in the past [4]. In this paper we describe a practically *no cost* speak-to-dial application built in-house and usable over the TCS internal VoIP (BUZZ). The implementation makes use of the speech recognition engine which comes preinstalled with most Microsoft Windows OS. The speak-to-dial service accepts (a) a continuous, free of format, speech query from the caller; (b) converts the speech into electronic text; (c) deciphers the person intended to be called and (d) connects the caller to the called person automatically. The speak-to-dial system is easy to use and can be used by anyone (speaker independent) without prior training. This paper describes the building and pilot implementation of the speak-to-dial system over the VOIP phone network. It also describes a method to enhance the performance of Microsoft's ASR engine, which actually uses acoustic models trained with American English, to (a) work for Indian accents and (b) to recognize Indian proper names by intelligent grammar modeling **Error! Reference source not found.**. We also provide preliminary experimental results to evaluate the performance of the speak-to-dial implementation.

1. INTRODUCTION

In many large organizations the most used channel for communication is the telephone for voice and emails for text. Organizations have adopted the use of VoIP (Voice over IP) especially to connect their offices spread geographically all over the globe. In Tata Consultancy Services (TCS [9]) analog and VoIP (also called BUZZ internally) telephones are a major means of communication between associates in and across offices. VoIP is not only economical but also improves ease of communication, connecting people and overall enhancing productivity to a great extent.

In any large organization there is a reasonable dynamics of people movement due to geographical office shifts. This necessitates the update of the telephone directory (name-number directory listing) frequently and in addition making it available to all the associates. However with the increasing number of interactions in growing work environment, knowledge of the person contact number is not easy to maintain. The usual practices in TCS (applicable to other organizations also) are one of the following

1. refer to an up-to-date organization wide VoIP directory listing [6]
2. to maintain a personal list of contacts (requires update especially if there is a change in the directory)
3. operator assistance or textual search directory (employee information service, like MasterFind)
4. remember most commonly dialed extensions (alright with a small list)

In the event, the person-number association is not known; it is a two step effort to connect to the required person. Namely, (i) obtain the contact number by any of the means 1, 2 or 3 mentioned above and (ii) dial the obtained number to connect. A natural (hence easier and convenient) way would be to have an automated service which allows one to call a common number and speak to be connected to a person and actually get connected directly without any human operator assistance. Speak-to-dial is a solution which allows people to speak to an automated system and get connected to the intended person without actually speaking to any human. A speak-to-dial system uses a speech recognition engine to convert the spoken request into electronic text; deciphers the intended person to be called and connect the caller to the called person automatically by searching the telephone directory.

This paper describes an implementation of a speak-to-dial service by TCS Innovations Lab - Mumbai. A ready to use speech recognition engine (comes pre installed on Windows XP machine) and telephony development interfaces has been used for building the speak-to-dial system. The implemented system has been tested over the BUZZ network and also on the regular telephone lines with success. The contributions of this paper are (a) development of a working speak-to-dial service for the TCS

TACTiCS – TCS Technical Architects' Conference

internal VoIP using readily available speech recognition engine using Microsoft's speech SDK and (b) using the speech engine, as it is (meaning, without any additional training of the acoustic models), to recognize Indian accents and Indian names by intelligently developing the grammar model **Error! Reference source not found.**. The paper is organized as follows: Section 2 describes the VoIP telephony in general and its implementation in TCS, Section 3 describes the implementation of the speak-to-dial architecture; Section 4 describes the challenges faced and innovative solutions adopted to over come the challenges; Section 5 gives some results of test setup and in Section 6 we conclude with the implementation cost, effect on productivity and future scope of this solution.

2. Voice over IP

Prior to the advent of internet interactive communications were only made by telephone at PSTN line cost, which was very expensive especially over long distances. With the advent of highly advanced network technology and computers being mass produced and available to consumers at a lower; people begun to communicate with new services like email, chat, etc. These services were extremely cheap and easy to use but lacked the personal touch provided by telephone. Therefore big corporations began to fiddle with the idea of allowing people to have real-time vocal communication and thus beginning a totally new chapter in internet history: VoIP [10].

VoIP stands for Voice over Internet Protocol and is a technology for transmitting ordinary telephone calls over the Internet using packet linked routes. VoIP is also referred to as IP telephony. VoIP involves the transmission of telephone calls over a data network like the Internet. VoIP telephone calls (voice) bypass the typical public-switched telephone network and transmit voice calls over a private network (the same network that carries web, e-mail and data traffic). In general Voice over Internet Protocol (VoIP) refers to the use of the Internet for making telephone calls. The main advantage for users of VoIP connections is that it avoids the tolls charged by ordinary telephone service and they generally only have to pay their usual (local) Internet connection charges regardless of where they are calling anywhere in the world.

Just like there are intra-organization telephone network served by a PBX (Private Branch Exchange) and connect the organization workers; there can be an intra-organization VoIP telephone network. The advantages of VoIP telephony are inherited here. TCS has such an internal VoIP facility called BUZZ. BUZZ is a faster, cheaper and simpler way to connect any user inside TCS organization[1]. It does not incur any STD charges as it uses the existing leased line TCS network to carry voice. Likewise, the network also gives a clearer and a faster way to communicate throughout the local BUZZ network. The other great feature of BUZZ is the roaming facility. The roaming facility enables user to take his extension wherever he travels. Any phone can be configured for any extension with user name and password (where the user name will be the extension number and password will be the security code associated with that number). A user can feed this information to any BUZZ phone and the phone will be configured for that specific extension. Apart from these features; call park, auto callback, directed call pickup, call forward, voice mail, call transfer and call conference [6] are the other facilities available on BUZZ.

3. Speak-to-dial IMPLEMENTATION

Figure 1 shows a block level diagram of the speak-to-dial service in general (not specific to VoIP) for a simplified telephone network (note that there could be several different PBX sitting between two phones – for sake of simplicity only one PBX is shown in Figure 1). The phones at different locations and the connection happen through the PBX. The speak-to-dial service is enabled on one of the phone lines[2] in the telephone network using a computer telephony card (CTI). A user dials that number and speaks (as he would speak to an operator), say, /Please connect me to Mister Aravind Shah/. The speak-to-dial uses the speech recognition engine to convert the acoustic speech into text, namely, "Please connect me to Mister Aravind Shah" (assuming that the speech engine is 100% accurate) and then processes the converted electronic text to derive the intended person to be called namely, "Aravind Shah"; looks up into the database to identify the phone number associated with Aravind Shah and then on confirmation automatically dials the number of Aravind Shah and connects the caller Aravind Shah and finishes the service.

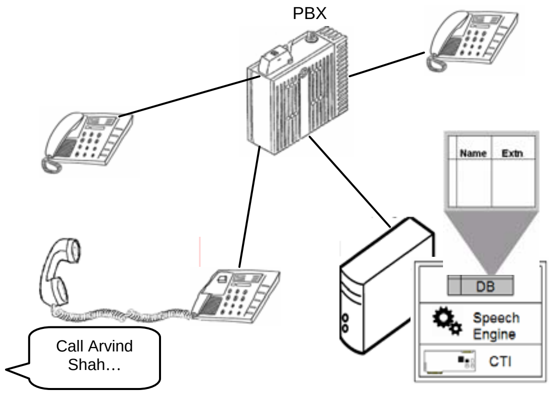


Figure 1. High level Block diagram of the Speak-to-Dial system.

In this form the speak-to-dial server hosts the Computer Telephony Card (CTI) card, the speech recognition engine, and the telephone directory database. The telephony part of the application with the CTI serves the call handling whereas the phone-number retrieval is done using a lookup in the database after the speech recognition application identifies the person to be called. In larger systems the CTI, speech recognition and databases may be hosted on different servers.

[1] BUZZ is a CUG (Closed User Group) [7] facility. Although it can be used for long distance calls, it is only inside the TCS group. Also it is totally separate and additional to the existing telephony network and as per TRAI regulations it is not connected to the PSTN.

[2] Say "*" which is a common key to get in touch with the operator

3.1 Call Flow

The call flow of the speak-to-dial application is shown in Figure 2. Speech is the interaction mode between the speak-to-dial service and the user. The user interaction is natural if it allows the users to speak in any way they like to get connected to the called party. For example, if the user has to speak the first name followed by the last name to use the service then the service becomes restricted in the sense that the user can intend the same thing (namely, speaking to Aravind Shah) but ask to be connected in several different ways, for example, (a) /Call Aravind Shah/, (b) /Could you please call Aravind Shah/, (c) /Aravind Shah/, (d) /Uhmm… can I get connected to uhmm… Aravind Shah/ etc. An unrestricted or easy to use system would not care in what form the user made the request and yet achieve the intent of connecting to Aravind Shah. The implemented speak-to-dial system allows user to speak naturally and freely. The most important aspect is the speech interaction which includes speech recognition as well as speech responses.

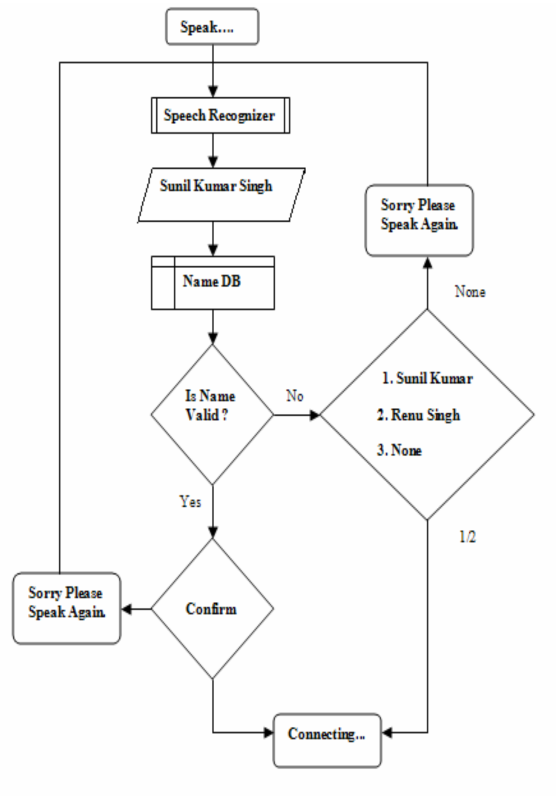


Figure 2. Call flow of the Speak-to-dial service.

3.2 Speech Recognition

The application allows the caller to speak freely and naturally, (in a broad sense, instead of just allowing the user to speak the name of the called party it allows the caller to speak naturally as if they were asking a human operator. The current implementation can understand speech queries derivable from Table 1. These include queries of the form – “Can you connect me to Sunil Kumar please”. Subsequent to speech recognition, the application recognizes the name from such a query and responds to the caller to confirm the name of the called party. The caller’s confirmation (a spoken /yes/ or /no/) is recognized and the call connect action is taken.

The speech recognition is carried out using Microsoft’s ASR (automatic speech recognition) engine. This engine comes with trained models for English language and uses the American English phoneme set and a built in pronunciation dictionary of English words. Recognizing English names, words and sentences is not very difficult. However a speak-to-dial application needs to recognize names (a fair number of them are Indian) and Indian names (or words) are not present in the lexicon supported by the speech engine. Additionally, the existing acoustic models (which come ready with the speech recognition engine) cannot be trained or modified to handle Indian accents. The speak-to-dial system was designed under these constraints. We achieve the name recognition by building an appropriate language model and a phonetic dictionary of names (using the American English phonemes). This is discussed in detail in a later section.

Microsoft provides a free to download and use SDK and a set of application programming interfaces called SAPI (Speech API) [3] for developing speech based applications. We use these to program the speech recognition using the Microsoft engine.

3.3 Speech Prompts and Responses

The speak-to-dial service communicates and responds to the caller through speech prompts and speech responses. For example, it confirms the name recognized with a prompt of the form - /Did you ask for Aravind Shah/, or prompts the caller to speak again if it could not recognize correctly, namely, /Sorry please speak again/. The implemented speak-to-dial service has a set of these predefined speech prompts and responses; which are used depending on the state in the call flow. The speech responses and acknowledgments of the speak-to-dial service play an important role in easy flow of the application (as is true for any speech based solution). These prompts must be simple to understand and short as well. Simple prompts and response increase the usability of the service. Whereas short prompts reduce the call connection time.

Microsoft provides a free TTS (text to speech) engine with male and female voices. Given a string of text the TTS engine speaks out the string in the selected voice. This can be used for the generating the speech responses for the speak-to-dial service. However the accent of the voices, pronunciation of Indian names (work on this has been described elsewhere [10]) by these voices and the clarity over the telephone channel is unacceptable. Pre recorded speech prompts is a possible solution to this problem and has been used in this implementation. The appropriate speech prompt (a wav file) is played depending on the call flow instead of generating using a TTS engine. Although it involves some efforts in recording these prompts, it gives a more natural interaction to the application. These prompts can be played back using either SAPI or TAPI (Telephony application programming interfaces) functions.

3.4 Telephone Interface and Call Control

The speak-to-dial server connects to PBX using the CTI card, which essentially is a voice modem which can handle voice calls over a telephone line and provide the streaming voice audio in digital format for processing. This digital voice audio can then be streamed to the speech recognition engine. The implemented speak-to-dial service uses a Dialogic D41 JCT CTI card **Error! Reference source not found.**. It has 4 telephony ports and can handle 4 lines simultaneously. The BUZZ VoIP line (the regular telephone line as well) can be directly connected to these ports.

The call answering, call transfer and call disconnect is programmed using the CTI API's. This is achieved using Microsoft's TAPI [8] (Telephony application programming interfaces) which comes as apart of the Microsoft Platform SDK.

Table 1. Structured grammar Rule char

Grammar					
[Begin Rule]			**Name Rule**		**[End Rule]**
[Filler]	**[Ask]**	**[Call]**	**[Salutation]**	**Name**	**[End]**
Hmmm *Ohhh* *Ummm* *Uhhh* *Ahhh*	*can you* *could you* *will you* *could you*	*[please] call* *[please]connect* *[please]connect [me] to*	*Mr.* *Miss* *Dr.* *Madam*	*N1* *[N2]* *[N3]*	*Please* *Madam*

4. Grammar and Name Dictionary

Any speech recognition engine has essentially two components, namely, (a) the acoustic recognition aided by the pronunciation dictionary and (b) the language model (also called speech grammar). The Microsoft ASR engine comes with acoustic model for the English language trained for English accent spoken in the US/UK and there is no provision to adapt or train[3] the acoustic models to cater to Indian accent. So the only control that is available to configure the overall performance of the speech recognition and hence the speak-to-dial application rests on intelligently modeling the speech grammar (language model) and building a pronunciation dictionary for Indian names [10].

4.1 Pronunciation Lexicon

For recognizing Indian name pronunciation we need to specify the pronunciation lexicon of the proper names to the speech recognition engine. SAPI provides a provision where user defined; alternate and multiple pronunciations for unknown and existing words can be specified. We use this provision to build a pronunciation lexicon of the names. This lexicon (dictionary) is used by the speech recognizer at runtime. A typical lexicon consists of name – pronunciation associations, the pronunciation being specified using the American English phoneme sets. For example:

JANARDAN	jh ah n aa r dh ax n
PINTO	p ih n t ow

Since the phoneme set is for American English, they must be correctly chosen when building the Indian name pronunciations. The chosen phonemes must be as close as possible to the phonetic sounds in the names. Adding multiple pronunciations can help improve the recognition of a word spoken by different speakers; or same speaker in different ways.

4.2 Speech Grammar Modeling

The language model is also known as the speech grammar modeling. It defines the probable sequence of words that the user might speak. SAPI provides two modes of grammar (a) dictation mode – where anything spoken is recognized using the standard in built language dictionary and (b) command and control mode – where there is a smaller defined set of words and sentences which is specified using an XML grammar file. We use the command and control mode since our recognition set is limited to sentences formed with small set of words (in Table 1) and names.

A structured speech grammar was developed for the speak-to-dial application. The grammar broadly consists of three parts.

1) The Begin and End rule have been designed to enable recognition of free speech query. These rules can further consist of one or several sub-rules which represent different parts in a query for example filler can handle utterances like uhmm, aah. In our implementation, these sub-rules are optional, to cater to users who may want to ask by just the name of the person they wish to talk against a complete statement query (/Anand Sharma/ instead of /Please connect me to Anand Sharma/.

2) The Name rule is the mandatory rule. In the grammar design we assume that a persons name can consist of a maximum of three name tokens (aka first, middle and the last name). However the terminology of first, middle and the last name is not used because not all names have all the three tokens. Additionally while referring to a person by name, (a) all of these tokens may not be used and (b) the sequence of these tokens may not be fixed. Each name is assumed to have a minimum of one and a maximum of three name tokens. The name rule defines a structured combination of these tokens

[3] The engine can be trained by individual users to create their profile. The engine will perform better for the user when his/her profile is loaded.

which can handle a name spoken in any combination (both in order and in number). This can be done in several ways; we have experimented with two

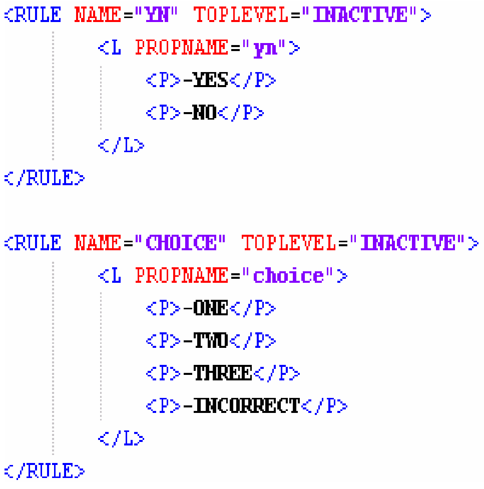
Type A: All the individual name tokens are represented as a set {N} and the name rule is of the form < N [N] [N] >. E.g. If {N} = {Sunil, Kumar, Kopparapu} then combinations like Sunil [Kumar] [Kopparapu], or Kumar [Sunil] [Kopparapu] or Kopparapu [Sunil] [Kumar] and other similar possibilities are allowed. (Note: [ ] indicates optional, may or may not be present in the spoken speech).

Type B: In this we have 3 sets {N1}, {N2}, {N3}. Where {N3} has all the names with three name tokens and each arranged in all possible orders of the three tokens, {N2} has all names with two name tokens in the possible four combinations for each full name and {N1} consists of all individual single name tokens. The name rule is

< N3> || < N2 [N1]> || < N1 [N2]> || < N1 [N1] [N1]>.

This kind of name rule has many repetitions of each name; however it directly states the name combinations in different orders. When names with more than one token are spoken, it reduces the chances of matching with other tokens, thus converging to single result, improving the accuracy.

3) The third part of the grammar (as shown in Figure 3) consists of rules to understand the confirmation from the caller like yes/no and choices one/two/three/incorrect in case of multiple results.



```
<RULE NAME="YN" TOPLEVEL="INACTIVE">
    <L PROPNAME="yn">
        <P>-YES</P>
        <P>-NO</P>
    </L>
</RULE>

<RULE NAME="CHOICE" TOPLEVEL="INACTIVE">
    <L PROPNAME="choice">
        <P>-ONE</P>
        <P>-TWO</P>
        <P>-THREE</P>
        <P>-INCORRECT</P>
    </L>
</RULE>
```

Figure 3. Confirmation rules from the actual Grammar file.

5. TEST RESULTS

The speak-to-dial service has been implemented and tested by about 10 different speakers within the lab. The name directory consisted of 36 names of the members of lab. A test bed of 77 spoken queries was collected from different and the preliminary results are shown in Table 2. Results show that the performance of the speak-to-dial system shows an overall performance accuracy of 54% when both grammar (B type) and pronunciation dictionary are used for a 0% recognition when none of then is used. When only the grammar is used the performance stands at 20% accuracy for type B grammar while it is 19% for type A. Just the use of lexicon does not help as seen from Table 2, where the performance is a partly 9% with pronunciation lexicon present and the grammar modeling (type A or type B) being absent. These results show that for an improved performance of the speak-to-dial system it is important not only to have a pronunciation lexicon but also a speech grammar.

Table 2. Test setup results. (A is grammar with < N [N] [N] > type name rule while B is grammar with < N3> || < N2 [N1]> || < N1 [N2]> || < N1 [N1] [N1] > type name rule.)

	Structured Grammar		**Dictation Grammar**
	A	**B**	
Without Lexicon	19%	20 %	0%
With Lexicon	48%	54%	9%

The above results were obtained using the Microsoft English Recognizer v5.1, which is essentially a speech recognizer for the desktop/microphone environment. The newer version of Microsoft's speech recognition engine, Microsoft English (U.S. Telephony) v7.0, which is designed for telephony environment, improves the performance for type A grammar from 48% to 64% and that of type B grammar from 54% to 62%.

6. CONCLUSION

The speak-to-dial service using Microsoft's SAPI can be practically employed on VoIP in an organization. It provides an easy solution to maintaining telephone contacts. This solution can be deployed on the analog telephone lines as well. It improves the ease of connectivity within the organization and thus in turn improves the productivity of the organization.

The cost of building this solution is practically zero especially considering the high cost of a commercial Name Dialer solution. This is because the speech recognition engines used by the in-house developed application are free and redistributable from Microsoft. The development tools for speech and telephony are also available easily and free of cost. Additionally, it uses the existing VoIP PBX system for dialing and connection. A full grown implementation can prove to be a practically low cost intra organization speak-to-dial solution for making BUZZ calls.

We have tested a prototype of the system for making calls internal to the TCS Innovations Lab – Mumbai. However for an actual deployment of the system and its use in the organization we need to (i) improve the accuracy of name recognition (ii) build the pronunciation lexicon for a large name list [approx. 3500 unique name tokens for TCS-Mumbai YP branch, 30000 unique name tokens for TCS-India] (iii) build speech prompts [synthesized or recorded] for these names (iv) build a scalable system which can handle large number of calls as well as multiple simultaneous phone calls and requests to recognizer. We have been working on automating ii & iii and working out i & iv for actual deployment.

We have demonstrated the feasibility of using a speech engine with acoustic models for one accent to effectively serve to address a completely different accent by tuning the speech grammar; additionally we also showed that the construction of a pronunciation lexicon enables the use of Microsoft ASR engine to recognize Indian names.

Speak-to-dial service can be enhanced for internal services like IDM helpdesk, security, canteen, travel which are often dialed numbers.

7. REFERENCES

[1] D41JCT Dialogic CTI card, http://www.dialogic.com/products/tdm_boards/media_processing/D41JCT_Boards.htm

[2] Laksmi Narayana, Sunil Kopparapu, “Semi-Automatic proper name lexicon Creation – An Optimization Approach”, ICON 2008, CDAC Pune, Dec 2008.

[3] Microsoft Speech API (SAPI). http://msdn.microsoft.com/en-us/library/ms723627(VS.85).aspx.

[4] Nortel Corporate Directory Dialer, www.nortel.com/products/04/st/collateral/nn106640.pdf; VoiceRite Name Dialer, http://www.voicerite.com/products/applications/namedialer/Name%20Dialer.pdf

[5] Sunil Kopparapu, “Voice Based Self Help System: User Experience Vs Accuracy”, The Fourth International Joint Conferences on Computer, Information, and Systems Sciences, and Engineering, 5-13, Dec 2008.

[6] TCS VoIP (BUZZ) on Ultimatix, https://www.ultimatix.net/

[7] Telecom Regulatory Authority of India, Consultation Paper No.4/ 2004, Consultation Paper On Application of principle of Non-discrimination in tariff schemes like CUG , VPN , F&F (Friends & Family) etc. February 13, 2004.

[8] Telephony Application Programming Interfaces , http://msdn.microsoft.com/en-us/library/ms734273(VS.85).aspx

[9] Tata Consultancy Services, http://www.tcs.com

[10] VoIP, http://en.wikipedia.org/wiki/Voice_over_Internet_Protocol