

TOWARDS BUILDING A MESSAGE RETRIEVAL FACILITY VIA TELEPHONE

Astha Goyal¹, Rochana Chaturvedi²

¹Department of Computer Science, Keshav Mahavidyalaya, University of Delhi

²Department of Computer Science, Keshav Mahavidyalaya, University of Delhi

Abstract

Email is one of the most popular services offered in the internet age. Therefore, providing service for all is a crucial competitive asset for any country. The objective of our research is to integrate technological innovation into a hybrid system, in order to build a more accessible system for weaker sections of society. This paper presents 'Telemail' system that enables a person to listen to the emails in her/his POP3 email account on a telephone thus replacing the conventional way of reading mails on the Internet. This software will serve the needs of people who don't have access to internet, are computer illiterate or blind.

Keywords—Telemail, Accessibility, Telephone, Deprived Users, ICT in rural India, voice based interfaces, Tool Prototype

1. INTRODUCTION

Communicating through emails is one of the most popular and fundamental activities on the internet. But still there are many people who can't afford the necessary means or others who can't read or see and are deprived of reaping the advantages that this technology could offer. Telemail is a tool we have conceptualized, where via integrating existing technologies, one would be able to hear their email over an ordinary telephone line. One only needs to call a number, enter the account number provided and through an interactive system, they can browse through the messages of their inbox.

In this research paper we primarily discuss implementation of the system. This is a hybrid tool and has been created using some of the available software. The implementation is divided into four phases. The first phase consists of programming an Interactive Voice Response System (IVRS) to make the computer an automated telephone system. Second phase comprises of maintaining a database for users and their messages. Mail is downloaded from POP3 email account. Extraction of various attributes of a message forms the third phase. Last phase consists of converting the text message into a .wav file through a text to speech converter. As far as hardware requirements are concerned, it consists of a modem whose purpose is to convert digital data into data over phone lines and vice versa. The hardware and the phases of software are integrated to achieve a prototype.

In our earlier work[I], we have presented the need for such a system, possible applications and the functionalities it must possess. In this paper we discuss the detailed implementation to develop a rapid prototype for this concept.

2. SYSTEM DESIGN

2.1 Architecture

The system architecture that is implemented for developing the system can be explained by the following diagrammatic representation.

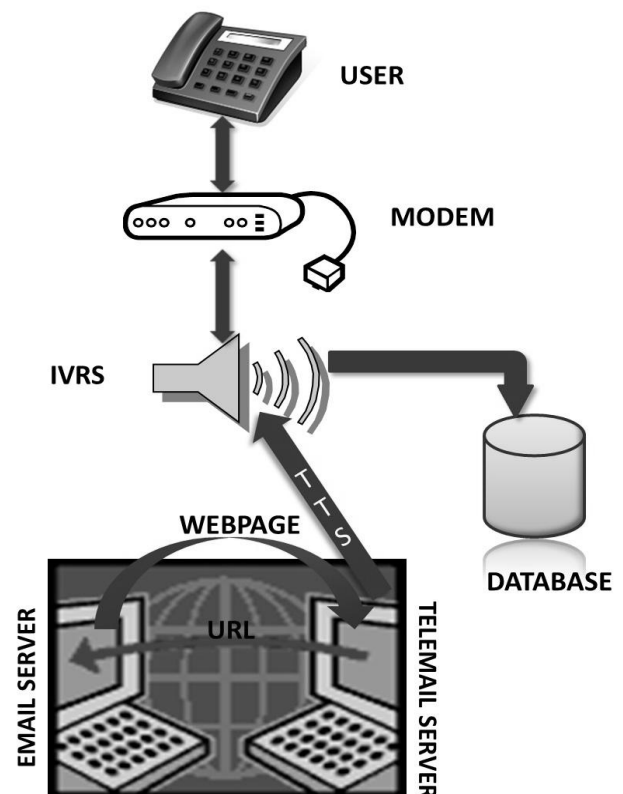


Fig 1: Functioning of the system

2.2 Use Case

The user calls the registered telephone number; the call goes through the connected system modem. Now the Interactive Voice Response System becomes active and communicates with the user by calling the modem functions. The voice messages are transmitted through telephone line to the user. The IVRS takes the key depressions as inputs provided by the user. Thus the user is provided an interactive menu system driven by use of voice commands and guided by user inputs. At the background of the IVRS there exists a module, which acts as an intermediate between IVRS and the database, where all permanent data of the system is stored. Another important part of the system accesses the database, takes the User Id's and Passwords of all the registered users, connects to the Internet and downloads all the messages from the users' Email box into a local directory for offline local access. After storage of all messages of the users a text to speech converter is used to convert the message file to a sound file format, which can be transmitted to the user via a telephone line by the IVRS system. The system is designed to be flexible, user-friendly, portable, and to support dynamic information.

3. SYSTEM REQUIREMENTS

3.1 Interactive Voice Response System (IVRS)

Since the prime objective of the system is to provide user with an interactive menu driven software environment, operable via a telephone and enable him to listen to the messages from his inbox, we require a software component with capabilities to act as an interface between the system and the telephone. This component sends out the sound files through the telephone to the user and store the data fed by the user for further manipulation.

IVRS also prompts the user for various checkmarks like informing the user about the nature of the data to be entered, how to enter the data in the correct format and updating the user with the status of data transaction. Besides these checkmarks this software should also prompt the user to either re-enter the data or end the session due to exceptions in the transactions, as per the situation.

3.1.1 IVRS used: Voice Guide version 5.0.5012

Voice guide software is used to serve as IVRS [II]. The software's trial version is a freeware and easily downloadable from www.voiceguide.com

- a) Minimum system requirements for running Voice Guide:
- Dialogic card, or CAPI compatible card, or Voice Modem, or other Telephonic Device.
 - Windows 95/98/ME/NT/2000/XP
 - Pentium 200MHz with 64MB RAM(Win 95/98/ME)
 - Pentium 400MHz with 128MB RAM(Win NT/2000/XP)
 - 40MB Hard Disk space

- b) Voice Guide Features:

- *GUI Interface:* The software has a script editor which implements the design through a graphical user interface environment in which the whole system is designed by placing modules on a voice guide script file by drag and drop method.
- *Result Variables:* Result variables allow modules to use the result of any of the previous modules in the script.
- *Paths:* Paths is the name given to the arrows which shows the transitions from one module to another.
- *Play:* The play module plays the sound file encoded in wav file format. The module waits for a key as input from the user. We can set the path taken by the module for any key pressed, in the paths tab. Hence the system can prompt the user with appropriate messages. The messages of the user are also played using this tab.
- *GetNumbers:* This module will play a sound file and save the number sequence entered by the caller. The user is given a specific amount of time to enter the data. This module helps in entering long continuous strings of numbers from the user, which further processed in various modules. This module also store the user's data in the system which can be used anywhere within the system.
- *Database Query:* The DB module will access and/or update information in databases. The information retrieved will then be available in the other parts of the script. Data can be retrieved and updated in the following data sources: Access, Excel, SQL Server, Oracle, MySQL, FoxPro, dBase and any ODBC data source.
- *Run Program:* This program will run a specified program. The program can be provided with parameters while calling it. This module helps to run peripheral programs independently or in synchronization with the IVRS system. Passing parameters also helps to provide process to process communication. Thus the IVRS can offload its work to programs which are capable of doing things outside its domain thus actuating the system's work domain. Result variables can be used while specifying the program's name and the parameters passed to the program.
- *Run VBScript:* This module will run a VBScript and helps in dynamic verification, modification and manipulation of data and extends the scope of Voice Guide to an even higher level by incorporating the VB Script functions in the system.
- *Hang up the Call:* This routine hangs up the current call. In case of normal completion of the session the Hangup Routine marks the culmination of the session. If the system expects that there is nothing left to interact with the user i.e. no input is required and no output is to be

produced, system may choose to terminate the session by moving to the Hangout Routine.

- *Say Numbers*: The Say Number module can speak up the specified numbers either in digits format or in number system format.

3.1.2 Integrating IVRS into the system:

Menu Implementation using Play Sound File, Get Numbers and Paths component:

- Play Sound Files*: The first step in the implementation of menu is to play sound files which hold the information about the nature of data to be entered, the correct format in which data is expected and also the messages indicating the session status.
- Get Numbers*: The next step is the get number module, which inputs the numbers from the users and stores them in the result variables.
- Paths*: The final step in the menu implementation is to decide the path on the basis of the data entered by the user and system parameters like time out of the module etc.

3.2 Data Store

This module deals with two parts:

- **Data Storage**: The data entered by the user is stored in the result variables associated with each module and can also be stored permanently in a database by use of database query module. This query module is used with an MS Access database, which supports SQL query format.
- **Data Manipulation**: Use of Vbscript enables us to retrieve, modify and update data stored in the database. This readily uses the result variables in the system and returns data back to the system by the use of result files. Various SQL statements can be used to retrieve data from the database.

Using peripheral programs by Run Program Module:

Peripheral programs can be invoked by a Run Program module. Parameters to the peripheral program are passed by the system itself like result variables.

3.3 Mail Download

Mail download is central to our Telemail system. The backend of the system required a robust mail enabled platform that could handle access protocols of a variety of complex mail message formats and data types. So we needed to provide an interface for accessing message stores of users and managing their electronic mails. In a nutshell, mail management performs the following two functions:

3.3.1 API Chosen: JavaMail API, JDBC Framework

- Mail retrieval and storage using Javamail API [III]: It allows service providers to formalize a standard interface to their messaging systems using the Java

programming language. Using this API, applications access message stores, and compose and send messages. It is composed of a set of abstract classes that models various pieces of the mail system. We can implement these abstract classes in term of our specific requirements. This API supports many different messaging system implementations-different message stores, message formats, and different message transports. It also provides a set of base classes and interfaces that define the API for client applications. Our software only needs to interact with the messaging system through these base classes and interfaces. JavaMail subclasses can expose additional messaging system features. For instance, the MimeMessage subclass exposes and implements common characteristics of an Internet mail message, as defined by RFC822 and MIME standards. We can also subclass JavaMail classes to provide the implementations of particular messaging systems such as IMAP4, POP3, SMTP [IV].

- Updating message database of registered users using JDBC framework: JDBC (Java Database Connectivity) is a standard API that is used to access databases regardless of the driver and the database product. MS Access is used for storing data, which is an ODBC (Open DataBase Connectivity) compliant database and JDBC-ODBC Bridge is used for connection to database.

3.3.2. Algorithm for mail download [V]

- Retrieve user id and passwords from database using account_no as provided by the user.
- Create a session object.
- Create a store object with POP3 access by calling getStore method on session object.
- Authenticate the user with userid and password through connect method on store object.
- Retrieve the 'inbox' folder by getFolder method on store object.
- Create an array of Message class objects to retrieve messages from the folder. The lower bound is decided by the number of unread messages.
- Obtain the attributes of the message via getForm, getSubject and getContentType methods.
- Update the database and personalized mailbox folder of the user with subjects, senders and message_nos.
- If the contentType is "text/plain", then the message is written to file in the personalized folder of the user. Otherwise if the message is of multipart type, then all BodyParts are retrieved iteratively and are appended to the file created for that particular message number in the personalized folder.
- After retrieving all the messages, close the folder.
- Close the store.

Following diagram represents this algorithm:

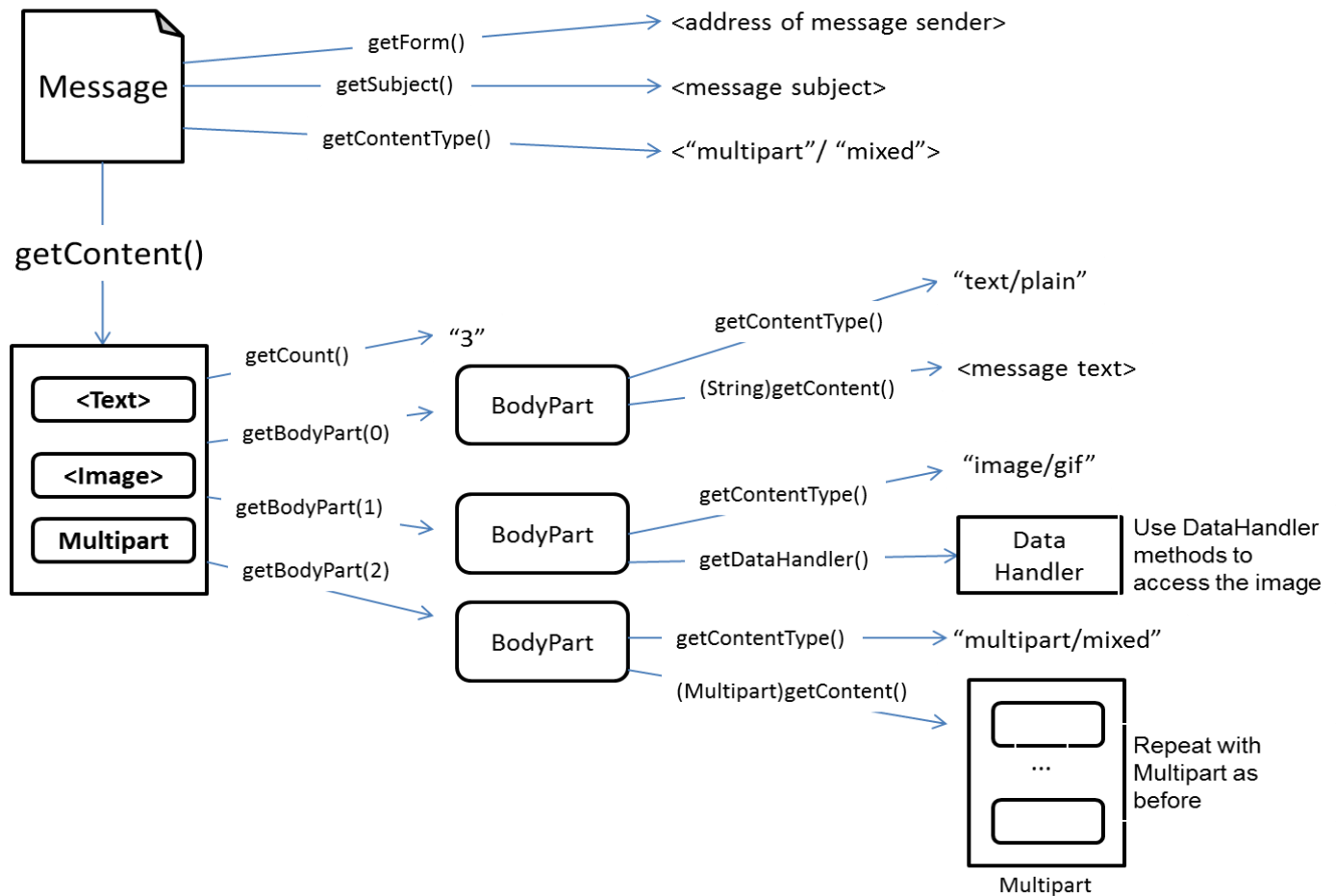


Fig 2: Extraction of Message Parts

3.3.3 POP3 Protocol Provider for JavaMail API

There are basically four protocols that are used with the API. These are: SMTP, POP, IMAP, MIME. POP stands for Post Office Protocol and its version 3, also known as POP3. A POP3 protocol provides access to a POP3 message store. It controls the connection between a POP3 email client and server where email is stored. This service uses the POP3 protocol for retrieving email from a mail server to a POP3 email client with an assigned port of 110. The POP3 provider is accessed through the JavaMail APIs by using the protocol name "pop3" or a URL of the form "pop3://user:password@host:port/INBOX". POP3 supports only a single folder named "INBOX".

YahooMail uses the POP3 protocol and we had tested our prototype on YahooMail but same can be configured for any mailbox. But Yahoo Mail disabled free access to its POP3 service on 24th April, 2002. YPOP's is an open source application that provides POP3 and SMTP access to Yahoo Mail. It is available on the Windows, Linux, Solaris and Mac platforms. This application emulates a POP3 server and enables popular email clients like Outlook, Netscape, Eudora, Mozilla, etc., to download email from Yahoo accounts. YPOP's is like a gateway and provides a POP3 server interface at one end to talk to email clients and an

HTTP client interface at the other which allows it to talk to Yahoo. YPOP's was inspired by a Perl Script called Fetch Yahoo written by Ravi Ramkisson. The host name to be used for YPOP's is 127.0.0.1.[VI]

3.3.4 Conversion of the Text Files to Wav Files

This was one of the most important modules of the entire project as the emails need to be heard through the phone line to the end user. Hence we needed to use a Text to Speech converter to convert the downloaded text files into wav files. TTS, or text-to-speech, is the digitized audio rendering of computer text into speech. TTS software can "read" text from a document, web page or e-book, generating synthesized speech through a computer's speakers.

TTS can convert text files into audio MP3 files that can be transferred to a portable MP3 player or CD-ROM. Since we are making an automated system i.e. there is no programmer working on the server, we needed a converter that would automatically select the text files and convert them into audio as and when the system requires it to. Another feature that was required from the converter was to convert sound files in such a format that could be transferred through the voice modem and on the phone line.

- a) *Text to speech converter used: Textsound 2.0* [VII]
It is a text to speech file converter where we can batch text documents to spoken audio files using TextSound Windows Edition or TextSound Command Line Edition. It supports different output formats, including .wav, and .mp3.

Features of Textsound:

- Batch converts text files to audio files, there's no text size limit.
- An intuitive interface makes Textsound extremely easy to use.
- Multiple ways to add text files, such as Drag-and-Drop and more.
- Fast access to comprehensive properties for selected conversions.
- Conversion speed is configurable for PC's best performance.
- Support different output file formats.
- The conversion list can be saved for later use.
- Textsound is compatible with almost all text-to-speech voices.
- Advanced users can perform conversions using command line.

- b) *Integrating Textsound with the system:*
When the user requests the IVRS to read the emails, YPOPS downloads the mails in the server and stores them in the text files. Then the Run Program module of the IVRS invokes Textsound which then converts these text files into wav files which are stored in the folder assigned to that particular user at the server. The output directories i.e. the locations from where the Textsound will select the text files to convert can be specified according to the requirements.

4. ROAD BLOCKS FACED

While developing this prototype a number of problems were faced, ranging from the non-functioning of certain services in a particular module to the incompatibility between the modules. So all the solution to these problems were properly researched and solved. We also tried to find the optimum solution to these problems. The problems with their solutions are as follows:

4.1 Interactive Voice Response System (IVRS)

The roadblock faced here was in the choice of IVM by NCH swift sound[VIII]. Though IVM had a wide range of functionalities, still we faced a number of problems with it. The main problem faced over here was in the connection with the database by the IVRS. This database connection constituted a very important part of the module through which the registered users' records were being added to the database. Hence IVM was abandoned and search for new IVRS was started which ended with VoiceGuide. The differences in IVM and Voice Guide have been described below:

Table: 1 Comparison Table: IVM vs Voice GUIDIE

Features	IVRS	
	IVM	Voice Guide
Multiple Line Support	Yes	Yes
Call Key select menus	Yes	Yes
Automatic Message forward by email	Yes	Yes
Call transfer	Yes	Yes
Call Simulator	Yes	
Automated Outbound Calls	Yes	Yes
Free Plugins	Yes	
Inbuilt ODBC Data Sources		Yes
Execution of VB script		Yes
COM Interface with ActiveX Server		Yes
Graphical Design Environment		Yes

4.2 Mail Download

We require that the emails should be downloaded from a particular user's email account via POP3 server and stored in the form of text files at the server. There are different ways in which we could achieve this. We started our search with Outlook Express and ended at JavaMail.

Microsoft Outlook Express is an e-mail and news client which can directly download emails from a user's account. However, the extraction of mails through Outlook Express was not possible as it stores the mails in its cabinet files whose format and the structure is not known. Hence we didn't have access to them. So we found an alternative to it that suits our requirements i.e. JavaMail.

4.3 POP3 Access

Javamail was an extremely good tool for downloading but a roadblock was found while downloading mails via POP3 server of the Yahoo Mail. The problem is accessing Yahoo Mail server was that it has now been turned into a paid service. Remedy to this roadblock was found via YPOPS software. This application is more like a gateway. It provides a POP3 server interface at one end to talk to email clients and an HTTP client (browser) interface at the other which allows it to talk to Yahoo i.e. it directs our server requests to it and in turn accesses the POP3 server of YahooMail without any cost or loss of facilities. Hence we were to achieve free POP3 access to the Yahoo mail and in turn able to download the emails.

5. RESULTS AND FUTURE WORK

A true system must comprise of many parts which are loosely coupled, in addition they must be cohesive enough so that new features can be added to it with relative ease. Although we have tried our best to include as much functionality in this prototype as possible, but still there is a

lot of work that can be done to improve this system further. Some of the features which we think can further enhance the functionality of this software and are possible to implement are listed below:

- *Notification to members when new mail arrives*- This can be achieved by calling the client at a predefined mobile number of sending an SMS as a notification.
- *Replying to Emails*- Replies to emails can be recorded in a sound file and sent back to the sender as attachment. A speech to text converter could also come in handy for this. Moreover the message composing mechanism for user needs to be perfected further, especially with respect to selection of recipients.
- *Multilingual Support*- This requires the support of Text to Speech converters of different languages.
- *Unification of multiple Email IDs*- If one person has 2 or more different Email IDs, then all his mails should be downloaded into a common account from where all the mails can be read.
- *Support of multiple mail servers*- Currently the prototype is configured and tested for Yahoo Mail Server. But these services can be extended to Hotmail and Gmail Servers also. Rediffmail server cannot be supported because POP access is charged in this case.

6. CONCLUSION

We have described Telemail, a tool for making email accessible to the so far deprived sections of society. It is still under evaluation in terms of costs and licencing but its free distribution will allow other users to add new functionalities. We hope that such a portable, widely available and flexible tool will benefit all of the target groups. Instead of developing similar tools in isolation, the research efforts of various scholars could be combined.

REFERENCES

- [1] R. Chaturvedi and A. Goyal, "Telemail- bridging the digital divide for rural hinterlands," IJOSR, .
- [2] Voice Guide Documentation. Electronic references. Retrieved from: <http://www.voiceguide.com>
- [3] JavaMail API Documentation. Electronic references. Retrieved from: <http://www.oracle.com/technetwork/java/index.html>
- [4] JavaMail FAQ. Electronic references. Retrieved from: <http://www.jguru.com/faq/server-side-development/javamail>
- [5] Fundamentals of JavaMail API. Electronic references. Retrieved from: <https://www6.software.ibm.com/developerworks/education/j-javamail/j-javamail-a4.pdf>
- [6] Free SMTP/POP3 access to Yahoo Mail. Electronic references. available from <http://www.yopsemail.com>
- [7] TextSound2.0. Electronic references. Retrieved from: <http://www.bytecool.com/textsnd.htm>
- [8] IVM by NCH Swift Sound Documentation. Electronic references. Retrieved from <http://www.nch.com.au/ivm/>