

Virtual Learning System (Miqra'ah) for Quran Recitations for Sighted and Blind Students

Samir A. Elsagheer Mohamed^{1,2}, Allam Shehata Hassanin³, Mohamed Taher Ben Othman¹

¹Computer Engineering Department, College of Computer, Qassim University, KSA

²Faculty of Engineering, Aswan University, Aswan, Egypt

³Electronic Research Institute, Giza, Egypt

Email: samhmd@qu.edu.sa, samirahmed@yahoo.com, almohager093@yahoo.com, maathaman@qu.edu.sa, mtothman@gmail.com

Received 1 December 2013; revised 2 January 2014; accepted 9 January 2014

Copyright © 2014 by authors and Scientific Research Publishing Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Quran has ten famous recitations and twenty different narrations. It is well known that the best way is to learn from qualified and authentic scientists (Sheikh's) in one or more of these narrations. Due to 1) the widespread of the Internet and the ease of use and availability of computers and the smart phones that enables access to the Internet; 2) the business of people hindering them to attend physical learning environments; and 3) the very few number of elder licensed scientists, we have developed a virtual learning system (Electronic Miqra'ah). Scientists can supervise remotely the registered students. Students (from different ages) can register from anywhere in the world given that they have Internet connection. Students can interact with the scientist in real time so that they can help them memorize (Tahfeez), guide them for error correction, and give them lectures or lessons through virtual learning rooms. The targeted groups of users can be non-blind people, blind people, manual-disabled people and illiterate people. We have developed this system such that it takes the commands via voice in addition to the normal inputs like mouse and keyboard. Users can dictate the commands to the system orally and the system recognizes the spoken phrases and executes them. We have developed an efficient speech recognition engine that is speaker independent and accent independent. The system administrators create several virtual learning rooms and register the licensed scientists. Administrators prepare a daily schedule for each room. Students can register to any of these rooms by pronouncing its name. Each student is allocated a portion of time where he/she can interact directly by voice with the scientist. Other students can listen to the current student's recitation and the error corrections, guidance or lessons from the scientists.

Keywords

Virtual Learning, Speech Recognition, Quranic Recitations, Learning Systems

1. Introduction

The Holy Quran [1] [2], also transliterated **Qur'an** or **Koran**, is the book literally meaning “the recitation” is the main religious text of Islam. It is the verbatim words of God (*Allah*). Quran was revealed through angel Gabriel from Almighty God to the prophet Muhammad (may peace be upon Him).

There are several sciences of the Holy Quran. One of them is: how to read, correctly according to specific rules (Tajweed). Quran is the basic science Muslims should learn and practice. Tajweed should be done according to rules of pronunciation, intonation, and caesuras established by the prophet Muhammad.

In addition to Tajweed, Quran memorization is of extreme importance for Muslims. Quran memorization and learning are among the best ways of worship that close to Almighty God because of their numerous benefits as stated in the Quran and Sunnah [1]. There are ten famous Quran recitations (a.k.a reading, or Qira'ah). In each recitation, there are two famous narrations (Rewayah). The most popular reading is that of Hafs on the authority of Asim. Similarly, each melodic passage centers on a single tone level, but the melodic contour and melodic passages are largely shaped by the reading rules, creating passages of different lengths whose temporal expansion is defined through caesuras [3].

These ten recitations consist of 7 Mutawatirs (a transmission which has independent chains of authorities so wide as to rule out the possibility of any error and on which there is consensus) and 3 Mashhurs (these are slightly less wide in their transmission, but still so wide as to make error highly unlikely) ways of reading [3].

Muslim scientists specialized in the recitation and the rest of Quran sciences learn the recitation from qualified and elder licensed scientists (Sheikh). Each one of these licensed scientist is linked in a transmission chain up to the prophet Muhammad. For more details about the Quran and its sciences, please refer to [2].

People seeking to learn the Quran are usually go to special schools: locations where a group of students are assembled and instructed by one qualified Muslim scholar. These special schools are known as Miqra'ah. The main purpose of Miqra'ah is for teaching Quran recitations and Tajweed as well as the rest of Quran Sciences.

Due to the lack of sufficient number of licensed Scientists, and the burden of the daily routines, very few students are going to Miqra'ah (which is usually located in Mosques).

With the widespread of Internet and its availability, the vast widespread of the smart phones having Internet access and the ease of use of these devices, it is important to develop a system emulating the physical Miqra'ah.

In this paper, we present a system that we have developed to serve as electronic Miqra'ah through the Internet. The system administrators register the licensed scientists into the system. They create several virtual learning rooms in the system and prepare a daily schedule for each room depending on the availability of the scientists. Each time slot in the schedule is assigned to one of the scientists. Using a program, scientists connect remotely from their homes or any location having Internet access to the system to the specified virtual learning room.

Students have access to the system through the Internet connects to the system and choose the virtual room that they want to join. After that, the scientist can teach them using voice broadcast (one-to-all). He can ask the students to recite and correct their recitations one by one. All the activities of the physical Miqra'ah are available in the system. However, people stay at home or work or in any place they have Internet connection and the client program.

The system is cross platforms compatible supporting Windows, Linux and Apple OS. The client application is also available for the most famous smart phones, namely those running Android and iOS. Thus, both the scientists and the students can access the system from their mobile phones in addition to the desktop and laptop computers.

The system includes very accurate speech recognition mechanism. Thus, students can dictate the commands by voice and the system can recognize them accurately and execute them. This feature allows the blind, illiterate, and manual-disabled people to make use of the system exactly like the non-disabled people who are able to use the mouse and the keyboard and read from the screen.

The rest of this paper is organized as follows. In Section II, we present the related works. We describe the Invented Speech Recognition Engine in Section III. Then, we present the used system for the voice communica-

tion over the Internet, namely Mumble VoIP in Section IV. We present the whole developed system in Section V. The conclusions and the future directions are given in VI.

2. Related Works

After a thorough and deep search, we could not find a system like the one that we have developed. When searching the Internet, we found several institutions claiming that they provide “electronic Miqra’ah”. Once examining them, we found the existing solutions are very modest consisting in using Skype or Paltalk to create a single unmanaged voice chat room.

The first electronic Miqra’Ah that we found is using Paltalk, by creating a room named “Riyadh Aljannah For Learning Quraan, for men” or “Riyadh Aljannah For Learning Quraan, for women”. They are located in All rooms→Middle East→Islam. Once the user join that room, the room admin (usually the Sheikh), can give the lesson, select any of the users and allow him to talk, all the others listen.

There are several problems with the solutions provided by Paltalk: 1) Annoying ads (usually containing offensive images/videos), 2) Only one room (no varieties), 3) Only one speaker, the rest only listen, 4) No announced schedule (people have to connect and wait until a lesson starts), 5) Not suitable for blind people nor manual-disabled people as they have to use the mouse and the keyboard and read what is on the screen.

The rest of the solutions consist in using Skype. The owner of the claimed electronic Miqra’Ah creates a user account on Skype and announces it. Anyone who wants to join that Miqrh’Ah has to add that user to his contact list and after approval from the owner, he can voice chat with that user when the owner is online.

Examples of the existing Skype accounts for that purpose are: *maqr2aonline.women*, *Maqrat. Quran, quraan 21*, and *almaqraa.alquranya*.

Skype shares the same drawbacks stated before for Paltalk. Furthermore, it has several drawbacks compared to the solution that we provide. Skype is really perfect for one-to-one call. However, for one-to-many calls, it is not suitable and the performance and the voice quality is very bad hindering it from being suitable for such task. In addition, there is no manageability with Skype, you cannot control who can talk now, for multi-party calls with Skype, at most three or four persons.

3. Speech Recognition Engine

We have invented a speech recognition [4]-[10] engine for the Arabic spoken phrases. It is based on the recently developed Google Speech API. It is speaker independent that means that any user can use it and it will recognize the spoken phrase accurately regardless of the specific personal feature of each user voice, without the need for training or retraining. It supports the Arabic language with several Arabic accents (Egypt, Saudi Arabia, Algeria, etc.) The accuracy is very high and it can reach up to 100% in the case when several simple rules are followed (speaking clearly and loudly, and disabling the recording enhancements of the microphone, and adjusting the recording level to 70% without boosting).

The mechanism is as follows (c.f. **Figure 1**):

- the user speaks a phrase, then we record his/her voice in a file
- Several sound processing are carried out (noise removal, filtering, silence removal, etc).
- The file encoding is then converted to “flac” encoding with the required sampling rate of 16,000 and mono channel.
- The resultant file is posted to “google” servers using the HTTP protocol with some parameters that we set for them. “Lang” parameter can be any of the supported languages: ar-Sa (Saudi Arabia), ar-Eg (Egyptian), etc. as in the following link.
<http://www.google.com/speech-api/v1/recognize?xjerr=1&lang=ar-Eg&client=chromium>
- Google Speech API servers convert the speech into text and return the text back to our application using JSON format.
- The accuracy of the recognized text by Google is very bad in many cases. For Arabic it is around 50%. Thus, we have added several post processing phases on the text to increase the accuracy.
- The post processing phase consists of using the approximate string searching and normalization within the expected spoken set of phrases.
- We have carried several experimentations of the post processing phase with 14 different approximate string matching algorithms [11]-[13], which are as follows:

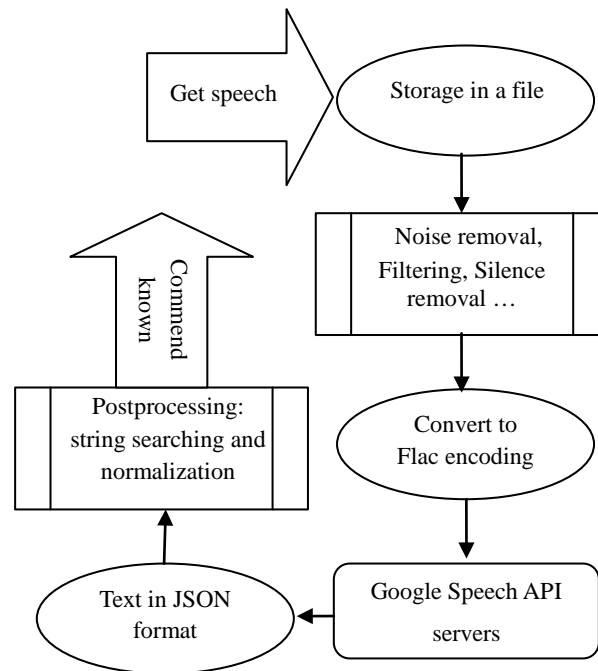


Figure 1. Block diagram of the speech recognition engine.

1. Smith Waterman;
2. Smith Waterman Gotoh;
3. Smith Waterman Gotoh Windowed Affine;
4. Q Grams Distance;
5. Needleman Wunch;
6. Monge Elkan;
7. Levenstein;
8. Jaro Winkler;
9. Jaccard Similarity;
10. Euclidean Distance;
11. Dice Similarity;
12. Cosine Similarity;
13. Block Distance;
14. Chapman Length Deviation;

- After applying each algorithm to recognize many texts recognized by “Google speech API” to the names of the names of Recitations and several categories of related and un related phrases, we found that the best algorithm that can be used to improve the accuracy for the phrasal string is “Levenstein” [13].

We have implemented this mechanism and then we have tested it on the recognition of the spoken control commands (e.g. start the program, end the program, I want to list to today schedule, I want to listen to tomorrow’s schedule, I want to listen to the help desk, I want to listen to the usage instructions or user manual, etc.) and the names of the channels (Miqra’ah) The accuracy is very promising, and it can reach up to 100% in all of the tests that we carried out given that the settings and the rules stated before are respected.

4. Voice over IP System (Mumble/Murmur)

After a thorough and intensive search for a suitable Internet communication system that enables the Scientists and the students to communicate with each other, we come up with several possibilities: Paltalk, Skype, Ventrilo, and Mumble. We have studied each one and the features provided by each of them, please refer to [Table 1](#).

Mumble [14] [15] is a Voice over IP (VoIP) software system primarily designed for use by gamers. Mumble uses a client-server architecture which allows users to talk to each other via the same server. It has a very simple

Table 1. Comparison between mumble, ventrilo and skype (source: [17]).

	Mumble	Ventrilo	Skype
Latency	Extremely low: communications happen in milliseconds, not seconds.	High: even low-ping servers have a 1 - 2 second delay due to the Ventrilo protocol	High: Skype is optimized for casual chatter, not realtime-when-it-counts communications.
Platform Support	Windows, Linux, and OS X	Windows, partial OS X support (some codecs not available), no Linux support	Windows, Linux, and OS X
Authentication	Certificate and password-based authentication. Strong certificate authentication ensures account security and user authenticity.	Password-based	Password-based
Voice Quality	Great-Mumble uses the cutting-edge CELT codec for maximum clarity.	Good-Ventrilo uses several codecs, such as Speex and GSM, though not all codecs are available on all platforms.	Average-Lower bitrate and high encryption overhead reduce quality.
Noise canceling	Automatic	<i>Not available</i>	Automatic
Push-to-talk	Available	Available	<i>Not available</i>
Linked channels	Available	<i>Not available</i>	<i>Not available</i>
User groups	Available	<i>Not available</i>	<i>Not available</i>
Text messages	Per-channel and per-user (private messages)	<i>Not available</i>	Per-conference and per-user
Administration	Done in the client. Add channels, manage users, and modify permissions in realtime.	Requires external web interface.	<i>Not available</i>
Access Control	Great-powerful ACLs allow for fine-grained control over all aspects of the server.	Coarse-users are either administrators or normal users.	<i>Not available</i>
Volume Attenuation	Mumble will automatically lower other system sounds when someone is talking. Play with music and game sound on-you'll never miss a message.	<i>Not available</i>	<i>Not available</i>
Customizability	Very high. Create custom layouts or skin the client to match your preferences.	<i>Not available</i>	<i>Not available</i>
Maximum users	Unlimited	8 for free server; unlimited with licensed server	10 on conference call
Comments & Channels	Rich text-include images and links as desired	Plain text	Not available
Scriptability	Great-Mumble is scriptable from many languages, including C++, Java, .NET, Python, PHP, Ruby, and Objective-C.	Not available	Not available

administrative interface and features high sound quality and low latency. Communications are encrypted to ensure user privacy. Mumble is free and open-source software. It is a cross-platform that means that the both client and server can run under Windows, Linux, and Mac iOS. In addition, there are client applications for the most common operating systems of the smart phones, namely Android and iOS. That means that the users can connect to the system from their Desktop, Laptop, Mobile phone, tablet, and notebooks, provided that they have Internet connection on these devices. We are going to describe the main features of Mumble in this Section.

Channel hierarchy: When installing the mumble server (a.k.a. murmur), the server creates on root channel, called root. System Admins can create as much channels as they wish under the root channel in a hierarchy. Users in each channel can talk together, and all can hear the voice of anyone who is talking. Admins can set permissions for each channel using the access control lists.

Sound quality: Mumble uses Opus [16], which is standardized by the Internet Engineering Task Force (IETF) as RFC 6716. Opus is the best for interactive speech and music transmission over the Internet. The sound quality is excellent compared to all the other existing software. It supports noise reduction and automatic gain control. It provides the lowest latency, resulting in faster communication. It also incorporates echo cancellation that permits the use of speakers as well as headphones without echo (Please see [Figures 2 and 3](#)).

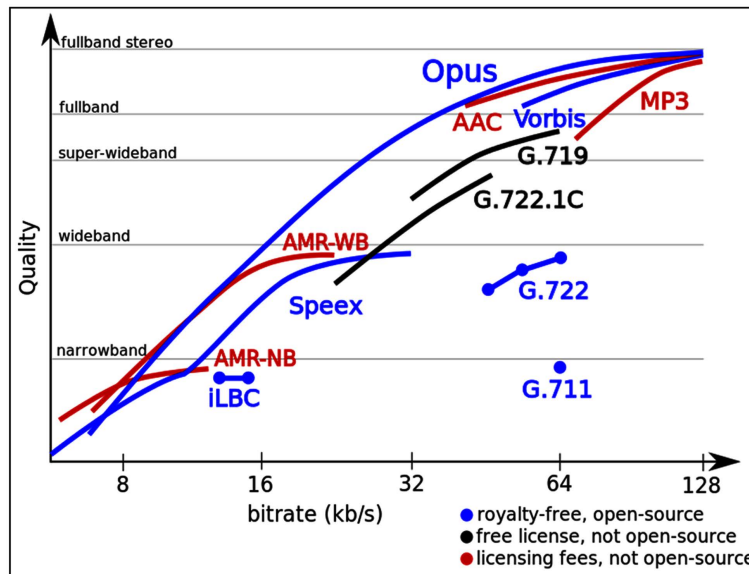


Figure 2. Comparison between the most known codecs for the quality against the bitrate (source: <http://opus-codec.org/comparison/>).

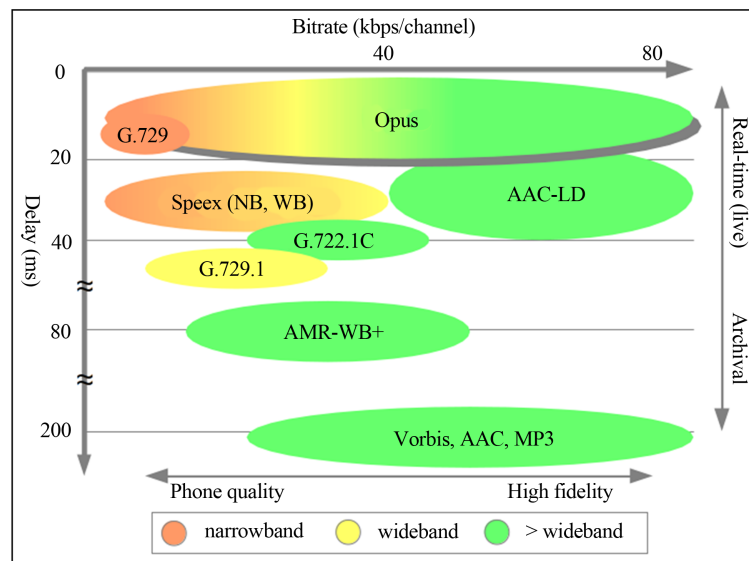


Figure 3. Bitrate/latency comparison (source: <http://opus-codec.org/comparison/>).

5. Developed System

After describing the essential components of the system namely the speech recognition engine and the voice over IP multiparty conferencing system (Mumble), let's present the whole system including the developed applications to provide the virtual learning environment (namely the electronic Miqr'ah). See Figures 3 and 4.

First, let's present the overall architecture of the system. The system consists of the following component:

- Murmur Server software
- Mumble client software programs depending on the platform
- Management and scheduling Web Application
- Speech recognition application for the blind, illiterate and manual-disabled people.
- Licensed Scientists attached to each channel according to the daily schedule
- System administrators



Figure 4. Overview of the system for a single channel (Miqra'ah).

- Students (non-blind, blind, illiterate and manual-disabled people)

A. User Interface

Two programs can be used depending on the user: non-blind who can use the mouse and keyboard, or people who are blind, manual-disabled or illiterate. In the case where the user can use the mouse and the keyboard, he can simply launch the client program from the Desktop, the laptop, the Smartphone, or the tablet. Then register into server and connect to the last channel (Miqra'ah learning room), he/she can move from one Miqra'ah to another by the mouse and the keyboard.

However, if the user is blind, or have hands disability, he/she has to use the other program. This program integrates the speech recognition engine described in Section III. This program has no GUI, it starts when the Windows start and runs in the background. It periodically play a beep sound, if the user says in Arabic "Start the program", the programs starts to interrogate the users by pronounce the available commands, and wait the user to dictate the required task. When the user says any of the available phrase to the microphone, the speech recognition engine recognizes the spoken phrase (command), and execute it:

The main menu of the program is as follows (no GUI, only voice announcement of the choices):

1) Listen to the titles (names) of the available channels (virtual Miqra'ah rooms)

In this case, the program fetches the available virtual Miqra'Ah names from the server and the corresponding sound files (prerecorded in the management web application: please refer to Section DV.D for more details). Then the program plays these files one after the other.

2) Listen to the schedule for today

In this case, the program fetches the schedule for today (Mecca time zone) prerecorded in the management server. The program plays the sound file for the Miqra'ah name, then the start time and the end time. It repeats until finishing the whole schedule. The time is pronounced automatically, for example, if it is from 05:35 am, the program will say the corresponding components by voice: "From five hour and thirty five minutes AM". Please refer to Section DV.D for more details.

3) Listen to the schedule for tomorrow

Exactly like the previous menu choice, but for tomorrow.

4) Join the last Miqra'ah room you have connected to.

In this case, the program launches the Mumble client application, connects the user to the server, it creates a specific username for the user containing the time and the real user name (the reason for that is to make a queue allowing the users entering the room to be in the top of the list, and those entering recently will be in the bottom of the list). The program makes the users joining the virtual room he/she were connected to the last use of the program.

5) Join a channel by pronouncing its name.

The system will ask the user to pronounce the name of the Miqra'ah room he/she wants to join. Note that, the user knows the channel names from the main menu. Then, the user says the name of the channel and the system recognize the required channel; then the program launches the client program (Mumble), and then connect to the voice server and makes the users to join the desired channel.

6) Stop or close the program

The system will close the Mumble client, and the user is removed from the users in the room he/she were connected to.

7) Listen to the user guide

The system plays back a recorded voice file explaining how to use the program efficiently as well as the rules of using the system.

The end users (students, teachers, and admins) can connect to the server using the Mumble client applications. For users who are able to use the mouse and the keyboard, they have to install the latest version of the Mumble client and install only the client application. Then they have to adjust the audio tuning wizard. Then, they have to connect to the server (required information is: the server IP or name, a real user name, and the password configured in the "*murmur.ini*"). Then the users can connect to the available learning rooms and select the one that he wants to join.

However, for those users who cannot use the mouse or the keyboard or who cannot read the text on the screen, they have to ask someone to install the software package that we have developed, and put a shortcut in the "Startup" Menu so that it starts with the system. A simple form asking them to fill the basic information will appear during the installation. The applications contain everything they need to connect to the server including the Mumble client. After that they will be able to make use of the application and the system independently without the assistant of any other person.

B. Server Management and Configuration

The whole system consists of: Mumble Server, Mumble Clients and the Management frontend. Here we provide the guide for *setting* up each of the components of the system.

The Mumble server is essential for doing most of the VoIP heavy works, organizing the channels, managing the ACL, and client connectivity. To set up the Mumble Server, one server machine is required, it can be dedicated, shared or Virtual Private Server in the cloud. The Supported OS is Windows Linux and Mac OS. The server must be connected to the Internet and have at least on public IP address. A DNS record for that IP address with the Server name is optional.

After preparing the server that will host the Mumble Server, the firewall has to be configured to open port 64,738. The bandwidth for the network connectivity for that server should be not be high. Mumble server uses the minimum bandwidth for with the best voice quality compared to the other solutions.

In Mumble documentations it is stated that at least 40kbps for each user given that all the users talks at the same time and all should here all (Mumble is all-to-all VoIP). However, we are not going to use it like that for two reasons: 1) in any of the learning room, the teacher should have the right to speak at any time, while only one students can talk and the rest of students should listen, otherwise managing the learning room would be very difficult. 2) if we allow all-to-all scenario, the bandwidth requirement will be very high.

Because of that, only the teacher and one of the students are allowed to talk at any given time in each learning room. This means we have two-to-all scenario. Using an intelligent mechanism coded in the client program, the user name of the student is prefixed with time he/she joined the lesson room. In this case, those who connects first will be in the top of the list, thus, the teacher will be able to know who has the turn. Once the turn of currently selected user is over, he/she mutes that user and un-mute the next user in the list.

The bandwidth requirement thus should be estimated as follows:

$$2 \times \# \text{ of rooms} \times \# \text{ of channels} * \text{quality_factor}$$

where the `quality_factor` ranges from 40 kbps to 110 kbps (highest quality). This value should be included in the configuration file of the Mumble Server (`murmur.ini`).

The latest Mumble package for the appropriate platform has to be downloaded and installed on the server (only install the server software and do not select the client software during the installation). Please follow the installation guide for the Mumble server available on Mumble site, and then change the following in the `murmur.ini` configuration file.

Password to join server. Very important and should match what is stored in the client application and known by the non-blind people

serverpassword = PFFFF

Maximum bandwidth (in bits per second) clients are allowed to send speech at (Set at your requirements, higher means better quality but requires higher server bandwidth).

bandwidth = 72,000

Maximum number of concurrent clients allowed.

Users = 1000

Amount of users with Opus support needed to force Opus usage, in percent. This is very important, if you will allow users having older versions of Mumble clients <v1.2.3, then set that value to 100. In this case, the quality of the voice may degrade as the server revert to the older codecs for all the users. However, if all the users will be forced to use the latest version, then it is safe to put that to 0. In this case, the quality of the voice will be excellent

0 = Always enable Opus, 100 = enable Opus if it's supported by all clients.

opusthreshold = 100

Regular expression used to validate channel names (change it to be as follows to allow Arabic names and some special characters).

channelname = [\\-|=\\w\\|\\#\\|\\[\\|\\{\\|\\}\\|\\(\\|\\)\\|\\@\\|\\|\\+]

Regular expression used to validate user names. (change it to be as follows to allow Arabic names and some special characters).

username = [\\-|=\\w\\|\\#\\|\\[\\|\\{\\|\\}\\|\\(\\|\\)\\|\\@\\|\\|*\\|\\W\\|\\+]

If this options is enabled, only clients which have a certificate are allowed to connect (very important to allow clients having old versions to connect. It is also important to allow the by-voice-controlled desktop application to operate properly).

certrequired = False

The rest of the options should be changed only in the case when you need to use them. Please note that, there is no need to use DBUS nor ICE (Internet Communication Engine) [18] [19].

After launching Mumble server, you have to put a link in the Startup menu so that it starts whenever windows starts.

Then you have to setup the SuperUser password using this simple command

`Murmur-supw <thepassword>`

After that from any other computer, install the client and connect to the server using the “SuperUser” account and the password you set in the previous step. Now it is time to create the channels (learning room).

Then create a group named “Teachers”; then edit the ACL permissions for this group by removing “Make channels”, and “Write ACL” permissions.

For each created channel, remove the “Speak”, “Write ACL”, “Make Channel”, “Make Temporary Channel”, “Mute/Deafen” and “Whisper” permissions for the “in” special group. Then add the teachers account to the Teacher group. And that is all for the Mumble Server configuration (Figure 5).

C. Client Configuration and Management

The end users (students, teachers, and admins) can connect to the server using the Mumble client applications (get it from [9]). For users who are able to use the mouse and the keyboard, they have to install the latest version of the Mumble client and install only the client application. Then they have to adjust the audio tuning wizard. Then, they have to connect to the server (required information are: the server IP or name, a real user name, and the password configured in the “`murmur.ini`”). Then the users can connect to the available learning rooms and select the one that he wants to join.

However, for those users who cannot use the mouse or the keyboard or who cannot read the text on the screen, they have to ask someone to install the software package that we have developed, and put a shortcut in the

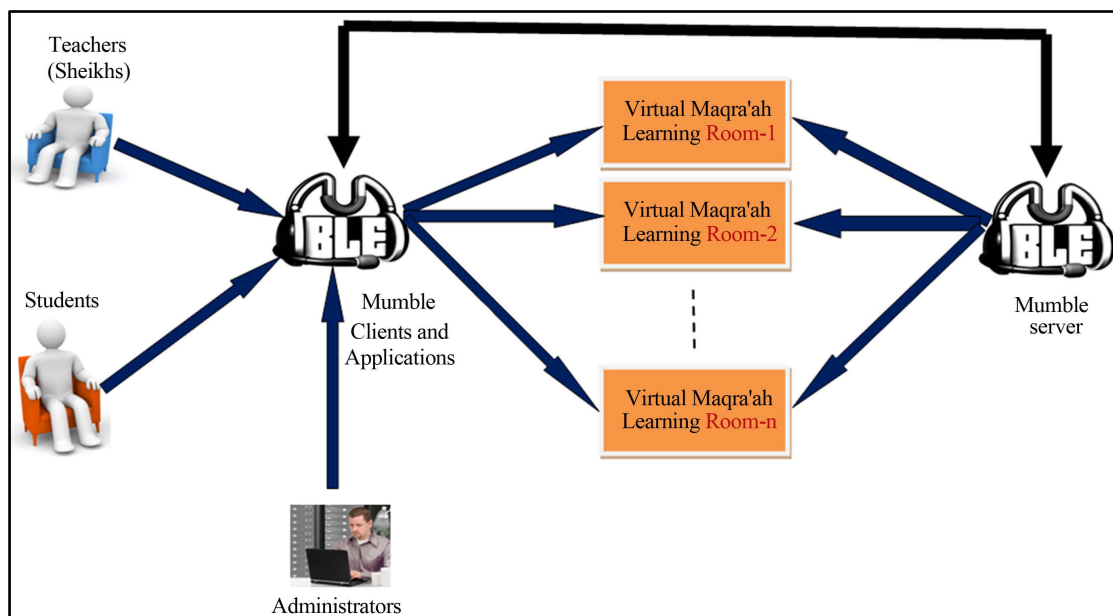


Figure 5. The system can have many virtual learning rooms, each one with a teacher and a number of students.

“Startup” Menu so that it starts with the system (can be obtained from <http://quranbyvoice.asites.org>). A simple form asking them to fill the basic information will appear during the installation. The application contains everything they need to connect to the server including the Mumble client. After that they will be able to make use of the application and the system independently without the assistance of any other person.

D. System Management

An integral part of the system is the system management application. We have developed a web application for managing the scheduling (timetabling), and assisting the client application for people with special needs.

This web application must be hosted in a webhosting server that is accessible publicly. The administrators of the web application have several duties to be done.

- 1) Create the list of the available Miqr’ah room names to be identical to those in the Mumble server. This will be used by the developed voice-controlled application.
- 2) For each Miqr’ah room name, they have to record a sound file in .wav format containing the name and upload that to the server.
- 3) They have to create the schedule of the available lessons every day. It is mandatory also to create the schedule of the next day.

All these duties are essential for the correct operation of the voice-controlled application. It is very important that the information stored in this web application be identical and coherent with that found in the Mumble server.

6. Conclusions and Future Directions

In this paper, we have presented a virtual learning system that we have developed to provide a real multi-virtual-room electronic Miqr’ah. The system consists of several components:

The system can be used by non-blind people. We have integrated a speech recognition engine that we have invented. Another application using that speech recognition engine is developed. It allows the blind, manual-disabled, and illiterate people to efficiently and easily use the system. They can have full control on the system using voice driven commands without having to use the mouse or the keyboard or to get help from other people.

The system incorporates a very performance, efficient, open-source, and reliable all-to-all voice chatting application, namely Mumble VoIP.

Licensed Sheikhs (scientists) as well as the students can be connected from their devices (desktop, laptops, and Smartphone’s), and conduct a lesson (Miqr’ah session), from their locations without having to go to schools or institutes.

The system is implemented and tested and it works perfectly. We believe that this system can be extended to be used for similar activities to conduct remote, distance and virtual learning sessions for people having blindness and manual handicap.

Funding

This work is sponsored by Noor ITC (www.nooritc.org), project no NRC1-29.

References

- [1] Islam City Website. <http://www.islamicity.com/mosque/quran/>
- [2] Islam Way Website. The Most Famous Broadcast Website about Islam. <http://en.islamway.net/>
- [3] Quran Readings, Wikipidia English Webpage. http://en.wikipedia.org/wiki/Qur%27an_recitation
- [4] Carnegie Mellon University (2010) Sphinx—Speech Recognition Toolkit. <http://cmusphinx.sourceforge.net/>
- [5] Lamere, P., Kwok, P., Gouvea, E. B., Raj, B., Singh, R., Walker, W. and Wolf, P. (2003) The CMU SPHINX-4 Speech Recognition System. *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, **46**, 37-51.
- [6] Abushariah, M.A.M., Ainon, R.N., *et al.* (2010) Natural Speaker-Independent Arabic Speech Recognition System Based on Hidden Markov Models Using Sphinx Tools. 2010 *International Conference on Computer and Communication Engineering (ICCCCE)*, Kuala Lumpur, 11-12 May 2010, 1-6.
- [7] Satori, H., Harti, M. and Chenfour, N. (2007) Introduction to Arabic speech recognition using Cmusphinx System. *Information and Communication Technologies International Symposium*, Fes, 3-5 July 2007, 4 Pages.
- [8] Hyassat, H. and Abu Zitar, R. (2008) Arabic Speech Recognition Using SPHINX Engine. *International Journal of Speech Technology*, **9**, 133-150.
- [9] Cambridge (2010) HTK—Hidden Markov Model Toolkit—Speech Recognition Toolkit. <http://htk.eng.cam.ac.uk/>
- [10] Young, S., Evermann, G., Gales, M., Hain, T., Kershaw, D., Liu, X., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V. and Woodland, P. (1996) The HTK Book. Cambridge University Press, Cambridge.
- [11] Shaikh, M., Memon, N. and Wiil, U.K. (2011) Extended Approximate String Matching Algorithms to Detect Name Aliases. 2011 *IEEE International Conference on Intelligence and Security Informatics (ISI)*, Beijing, 10-12 July 2011, 216-219.
- [12] Navarro, G. (2001) A Guided Tour to Approximate String Matching. *ACM Computing Surveys*, **33**, 31-88. <http://dx.doi.org/10.1145/375360.375365>
- [13] Gusfield, D. (1997) Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology. Cambridge University Press, Cambridge, 263-264. <http://dx.doi.org/10.1017/CBO9780511574931>
- [14] Mumble: An Open Source, Low-Latency, High Quality Voice Chat Software. <http://mumble.sourceforge.net/>
- [15] Rothbucher, M., Habigt, T., Feldmaier, J. and Diepold, K. (2010) Integrating a HRTF-Based Sound Synthesis System into Mumble. 2010 *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 4-6 October 2010, 24-28. <http://dx.doi.org/10.1109/MMSP.2010.5661988>
- [16] Valin, J.M., Vos, K. and Terriberry, T. (2012) Definition of the Opus Audio Codec. IETF RFC 6716.
- [17] Comparison between Mumble, Skype, and Ventrilo. <https://mmo-mumble.com/help/comparisonchart>
- [18] Henning, M. (2004) A New Approach to Object-Oriented Middleware. *IEEE Internet Computing*, **8**, 66-75. <http://dx.doi.org/10.1109/MIC.2004.1260706>
- [19] Khan, S., Qureshi, K. and Rashid, H. (2010) Performance Comparison of ICE, HORB, CORBA and Dot NET Remoting Middleware Technologies. *International Journal of Computer Applications*, **3**, 15-18.