CMPE 492 PROJECT REPORT

VIDEO CONFERENCING SYSTEM WITH VOICE RECOGNITION

by

Enes Ataberk
Hakan Selvi

Submitted to the Department of Computer Engineering
in partial fulfillment of the requirements
for the degree of
Bachelor of Science
in
Computer Engineering

Boğaziçi University
June 2010
# TABLE OF CONTENTS

TABLE OF CONTENTS ........................................................................................................... 2

ABSTRACT ............................................................................................................................... 3

1. INTRODUCTION .................................................................................................................. 4

2. PREVIOUS WORK ............................................................................................................... 4

3. SKYPE ................................................................................................................................ 5
   3.1 SKYPE API .................................................................................................................... 5
   3.2 SKYPE API REFERENCE ............................................................................................... 5
   3.3 SKYPE COMMANDS .................................................................................................... 7

4. VOICE RECOGNITION ......................................................................................................... 9
   4.1 VOICE RECOGNITION FUNCTIONS .......................................................................... 10

5. FLASH APPLICATION ......................................................................................................... 11

6. GOALS ............................................................................................................................... 12

7. BUILDING AND RUNNING THE PROGRAM .................................................................. 12

8. CONCLUSION AND FUTURE WORK ............................................................................. 13

9. REFERENCES .................................................................................................................... 14
**ABSTRACT**

Project Name : Video Conference System With Voice Recognition

Project Team : Enes Ataberk, Hakan Selvi

Term : Spring 2010

Keywords : Skype API Reference, Flex Builder, Flash Media Server

Summary : This project is composed of two parts. First part is C# project which uses Skype APIs and second part uses flash to make interactive video conference. In fact C# project is main part of our application and all user interfaces are located in this part. Flash application is a background service of our whole application and it uses Flash media server at the background.
1. Introduction

This document is written for Cmpe 492 Senior Project course given by Computer Engineering Department at Boğaziçi University. This project’s aim is to make a very user friendly interface for old people and patients to make them easily start video conference with his/her relatives. This is a kind of social project and will be useful for not only elders and patients but also their relatives who want to contact his/her elders.

By means of voice recognition elderly people can start a voice or video conference by only saying his/her doctor’s or child’s name whoever he/she wants to talk. And by means of touch screen clicking on the picture of the relative a video or voice call can be started and they can call anyone’s phone number by the aid of this project.

2. Previous Work

This project is a part of elder’s and patients’ house project. We go to the Kandilli Campus and see the lab then we have introduced the existing project and our project. We have informed about that our application will be which part of the big project. After we see the whole picture we have made a design of our project and present it to the project assistant. We have created a user friendly design for old people with large buttons, texts with large fonts, big camera panels for video conference and easily understandable user interface due to the computer illiteracy of elderly people and patients. After presentation to the teaching assistant, she made some suggestions and feedback about our project’s design and structure of our project. Then we changed the design, relative structures, and start to implement the project.
3. SKYPE

When the design issue has been handled we started to search how to implement our project. The project description suggests that Skype APIs can be used for video and voice conferencing. Then we download and search how Skype do the video and voice conferencing, how a user can start a call, send a message and call a cell phone. After we were familiar with Skype, we think that Skype APIs will be sufficient for our application.

3.1. Skype API

When we decided that Skype will be our milestone, we started to search how to use Skype APIs. Previously we decided that our application will be implemented on C# language because we are good at this language and user interface’s design properties can be prepared easily with C#. For this reason, we searched Skype APIs compatible with C# language. We found some Skype APIs compatible with Java, C++ and at last we found C# which is our preference.

3.2. Skype API Reference

After we found Skype APIs compatible with C# language we downloaded some necessary libraries for using some Skype commands. These are Skype Client and Skype Control libraries.

The api reference we found lists lots of Skype commands that can be used in C#. These are sending a message to a person, starting and finishing video and voice calls, making
voice conference, looking that the user is online or offline, looking previous message history, getting current call id etc.

By using Skype Control library and its functions we can use some sample functionalities of Skype in our own application. For example, we sent a message to one of our Skype contacts in C# application and then get the information about a user is online or offline.

To use Skype Control library we add this reference to our project definition:

```csharp
Using SkypeControl;
```

Then we add an attach handler and response handler events of Skype to control the Skype responses of any Skype events and attach the program to the Skype. Firstly, we attached the program to Skype and then an event fires. After that event fires, we can use Skype functionalities like sending message, starting call etc. which are previously defined. These are the events of SkpyeResponse and SkpyeAttach:

```csharp
aSkype.SkypeAttach += new SkypeAttachHandler(aSkype_SkypeAttach);
```

```csharp
aSkype.SkypeResponse += new SkypeResponseHandler(aSkype_SkypeResponse);
```

We use these events in our application by using these functions:

```csharp
void aSkype_SkypeAttach(object theSender, SkypeAttachEventArgs theEventArgs)
{
    theEventArgs.AttachStatus
}
```

SkypeAttachEventArgs is an event that fires when our program is connected to Skype. When we start the program Skpye asks permission for the program. After giving permission to our program then this event fires and we know that our program is connected to Skype. When the program is connected to Skype theEventArgs. AttachStatus text changes to
“Success”. Then we control that whether or not its text is exactly “Success” if it is then start our application. If not then warn the user it is not connected to Skype.

We use Skype Response event like this:

```csharp
void aSkype_SkypeResponse(object theSender, SkypeResponseEventArgs theEventArgs)
{
    theEventArgs.Response
}
```

Here SkypeResponseEventArgs fires when we use any Skype function in our application. By the aid of this event handler, we can develop project simply because we can find possible errors with these event handlers. We get the response of that event by looking at theEventArgs. For example, response text, if it is “INPROGRESS”, then we understand that there is an ongoing call. If it is “OFFLINE” then the destination user is currently offline and we do not send message or start voice call to that user.

After connecting to Skype we should use its functionalities. First of all, we define a SkypeProxy class and use its “Command” function to make calls and send messages. Here is the Skype Proxy class definition:

```csharp
SkypeProxy aSkype = new SkypeProxy();
```

And then with aSkype.Command(here is the functionality of skype that we use) command we use lots of functionalities of Skype.

### 3.3. Skype Commands

Here are the Skype commands that we use in our project and some explanations about them.
• aSkype.Command("ALTER CALL " + conferenceID + " HANGUP");

In this command, the application finishes current call which has call id or conferenceID. In Skype operations about calls are made by using ids of calls.

• aSkype.Command("CALL hakanselvi");

In this command, we call somebody by using his/her Skype name. For example, here we call the user hakanselvi who has a Skype account in this name.

• aSkype.Command("CALL +905058299553");

In this command, we call somebody by using his/her telephone number. For example, here we call the number +905058299553.

• aSkype.Command("SET CALL " + conferenceID + " STATUS ONHOLD");

Here we set the call with call id conferenceID status ONHOLD. We make this for voice conferencing in Skype. Because making a voice conference first a call must be started. After that its status set to onhold then the program

• aSkype.Command("SET CALL " + callID + " JOIN_CONFERENCE " + conferenceID + ")

Here we connect two users and our own skype user to a voice conference. As mentioned before conferenceID status set hold and a new call callID started. Then by this command this new call is joined to the holded call and a voice conference starts.
Here we send message to related user “p1name” for video conferencing. In this message, we define a person who is sent the message,”p1name”, and we send him/her a link to access the video conference on the web.

4. Voice Recognition

As mentioned before, voice recognition is a part of our project. It is used especially in emergent and abnormal situations. For example, when an old person or patient falls and needs help then he can say “call doctor” and our program call the doctor. If the doctor responds this request then a connection starts. Voice recognition is not only for emergencies. If the old person doesn’t want to press a button then he can say the name of the person which he/she is saved previously then the program starts a call.

We searched voice recognition code on the internet and found many programs like sphinx, komutanlar. However, we don’t want to a third party program so we need a C# voice recognizer or something like that. Then we find a speech recognition library of C#. It has simple functions and recognizes only English words. In addition, its recognition engine cannot discriminate words with similar pronunciations. But this is not our main issue about our application. So, we decided that C#’s own library will be sufficient for our application.

Since we make a voice conference and video conference we separate the voices recognize two conference types. For example, if the user wants to make a voice call with his son he/she can say “call my son” and a Skype request is sent to the user’s son. If the son accepts this call then a voice call starts. If the old person wants to make voice conference then after he/she says “call my son” he/she should say for example “call my daughter” and then the user, his son and his daughter will be in a voice conference.
Similarly, if the old person wants to make video conference then he should say “video my son”. Then a link sent to his son with Skype and if the son clicks on that link then a web cam conference starts. Video conference includes voice conference also which means voices also sent as videos on web cam. If the old person wants to connect his daughter to the video conference then he should say “video my daughter”. Then same link also sent to his daughter and if the daughter clicks on that link then a 3-person video conference starts.

4.1. **Voice Recognition Functions**

As mentioned before we use C# voice recognition library named with “System.Speech”. Then we include it to our project as:

```csharp
using System.Speech.Recognition;
```

Here I will define the functions of Speech library one by one

- **private SpeechRecognitionEngine recognizer = new SpeechRecognitionEngine();**
  
  In this code, we define the speech engine of C#.

- **private void LoadGrammars()**
  
  In this function, we define the set of strings that the language will recognize. Then we create a grammar builder and set its values to these strings. After that we create a grammar and use recognizer.LoadGrammar() function to load the grammar.

- **private void StartRecognition()**
  
  In this function, we define the speech recognition and speech reject events.
This event fires when a speech is detected by microphone but not recognized yet. It gets audio as input but does not compare it with the predefined grammar yet.

5. **Flash Application**

We make our video conference by using Flash (Flex Builder) and Flash Media Server (FMS).

Firstly, we search how to make video conference with Skype APIs and we arrive at a conclusion that current Skype API does not allow video conference with more than 2 people. Therefore, we searched how we can make a video conference by using C#’s own libraries because we did not use a third party program again. Then we found some example C# programs that get the web cam video and show it on the screen. However, these applications do not connect more than two users.

After that, we decided to use a third party application. By searching on the internet, we found some web sites that make video conferences. These web sites work as follows: First, a user enters the website and opens a room for video conferencing. Then the other users enter that room and see the previous users’ web cam videos on the screen. By these web sites, at most 16 people can make a video conference.

When we search background work of these web sites see that most of them are Flash projects and uses Flash Media Server (FMS) database. This database is especially for video and voice files. Then it is the best for our video conferencing system.
Any user can reach the database by IP number of the host machine. When they reach the database the video files from the web cam are saved to the database and the other web cam video files are loaded from the database and shown on the screen. This is the basic layout of our Flash Media Server.

The usage of Flex Builder tool is a little bit complex. The details of this work can be found in installation document.

6. Goals

As mentioned before, the goal of this project is making a video/voice conference with speech recognition. This is also a social project because it connects the old person to its relatives by seeing them on the screen and hearing their voices. The other goal of this project is it can be regarded as an emergency alarm. Because when the old person for example had a heart attack then he can call the doctor easily only saying “call doctor”. For this reason, this project may save live.

7. Building and Running the Program

Since our project is written in C# language to run the program someone needs a C# compiler. To make a video conference the one needs flash plugin embedded in the web browser and a web cam or another cam device on the computer. And for voice recognition a microphone and a speaker is needed.

Before running of the project with C# compiler the user’s Skype account must be opened. After running, the user’s must give permission to the program to reach Skype.
FMS is needed on a machine. This machine may be one of the video conferencing machines or another machine that is connected to the network. So FMS installed machine do not have to make a video or voice conference. This machine may be a third party machine but it must be connected to the same network with video conferencing machines.

8. Conclusion and Future Work

Since our FMS works for only local network the future work may be publishing it to internet and reach the people from other networks. By making this, we can reach more people but this may cause some speed problems. However, these problems can be solved by buying high-speed servers that are (more expensive).

Another future work may be on voice recognition system. Since we use C#'s own library and it has a poor voice recognition engine. A more powerful speech engine can be used for discriminating the words that have similar pronounciations.

Our program is for only voice/video conferencing systems. A chatting system can be added to our application. However, a question comes into the minds that whether an old person can use chatting system efficiently or not?
REFERENCES


