SPEECH TO TEXT CONVERTER APPLICATION

Guided By:

Prof. Mrudang Mehta

Prepared By:

Mitul Shah (CE-104)

Ruchik Shah (CE - 107)

Introduction

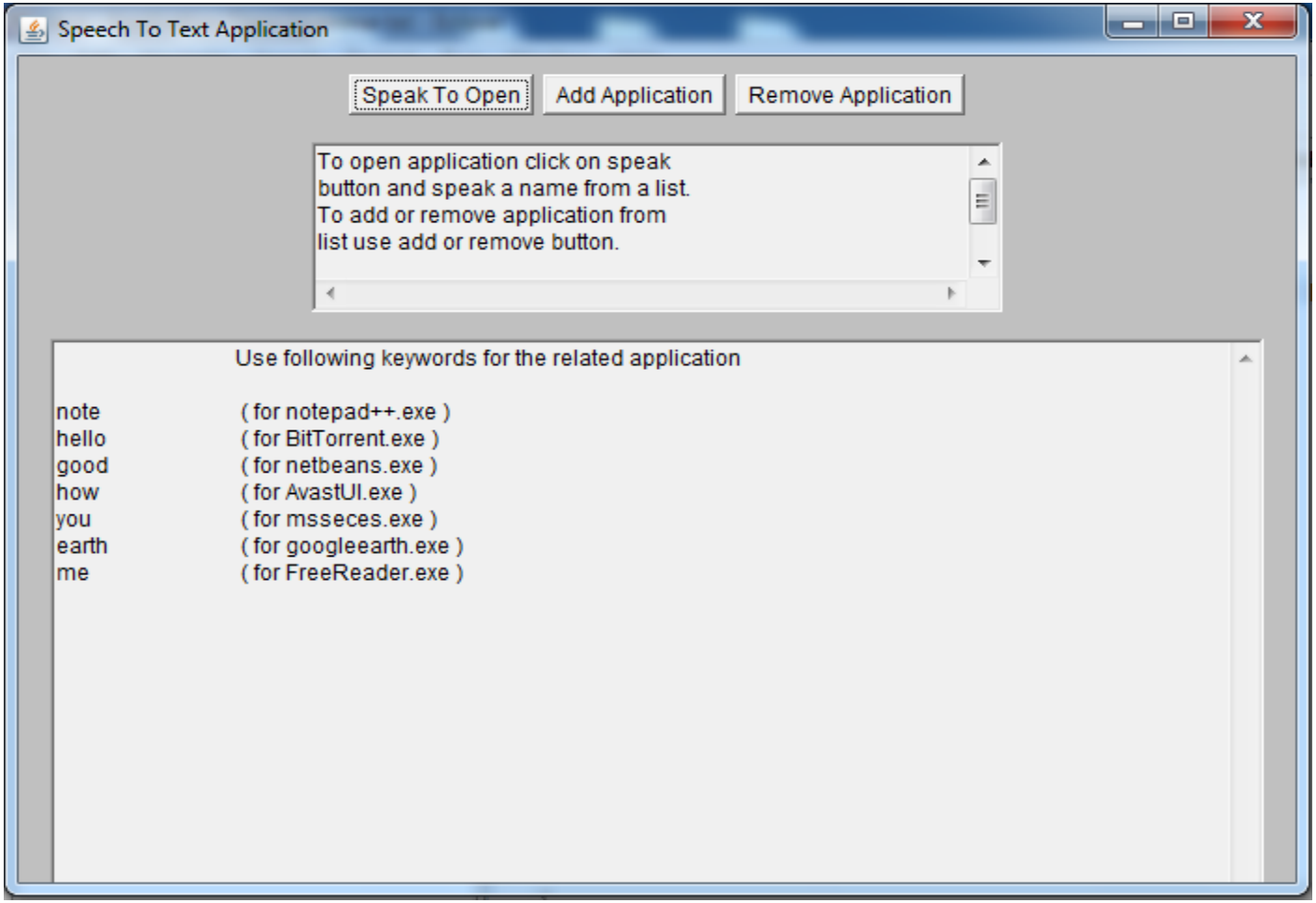
- The main functioning of our system is to execute any application on host machine without using keyboard or mouse!
- It basically works on speech technology.
- The user have to just speak name of application. And if that application is added to database before it starts executing.
- It mainly reduces human efforts and saves time also.

Technologies and Tools used

- Java Speech API
- Microsoft Access as database
- Cloudgarden implementation of JSAPI
- Microsoft SAPI v5.1 (Recognizer)
- Eclipse IDE
- JSGF

Basic idea

- When you run our application it looks like this :

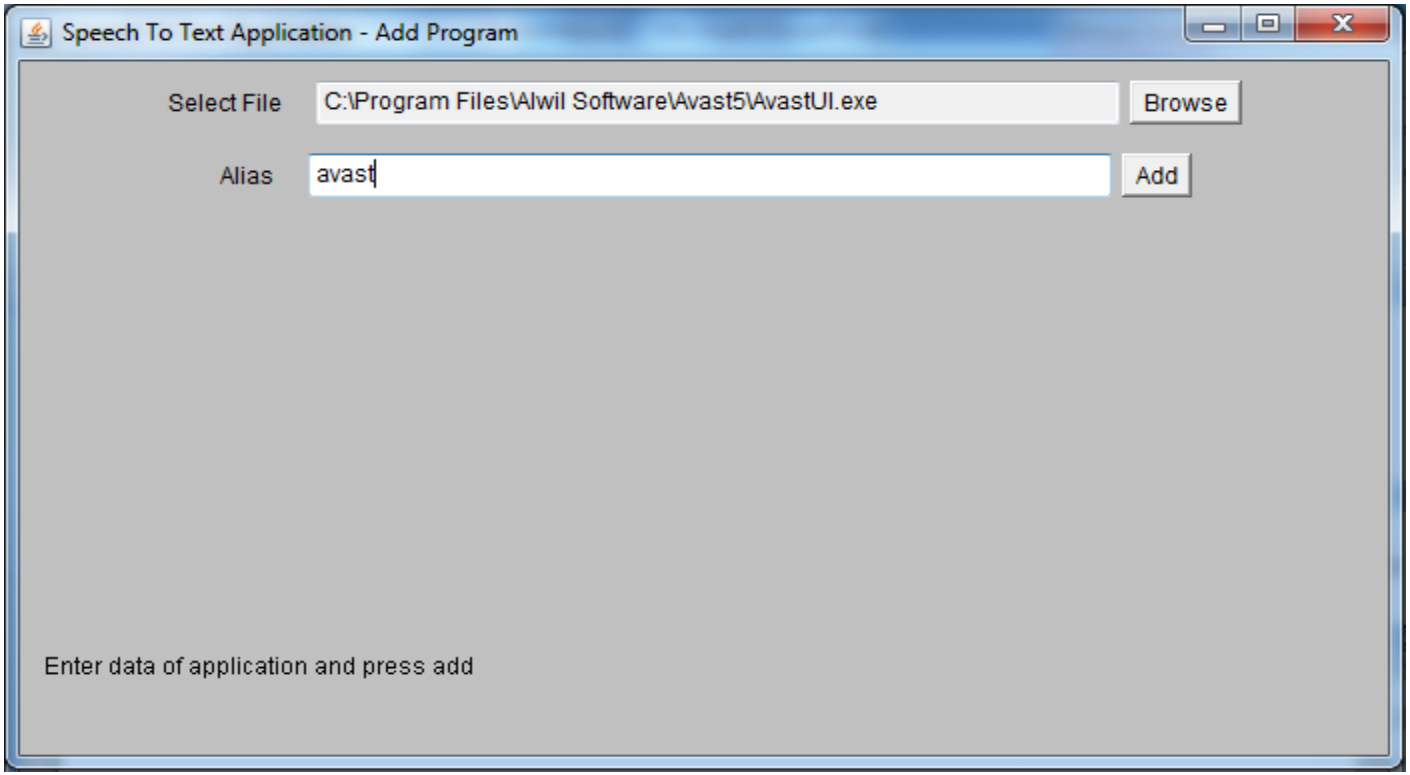


Basic idea

- In the GUI shown there are 3 main parts
  - Buttons to perform certain actions
  - Text area to guide the user
  - List of application in the database added before
- Now to start executing any application user first have to press “Speak to Open”. And than speak to open any listed application.
- If user wants to add or remove any application he can do that by pressing according buttons.

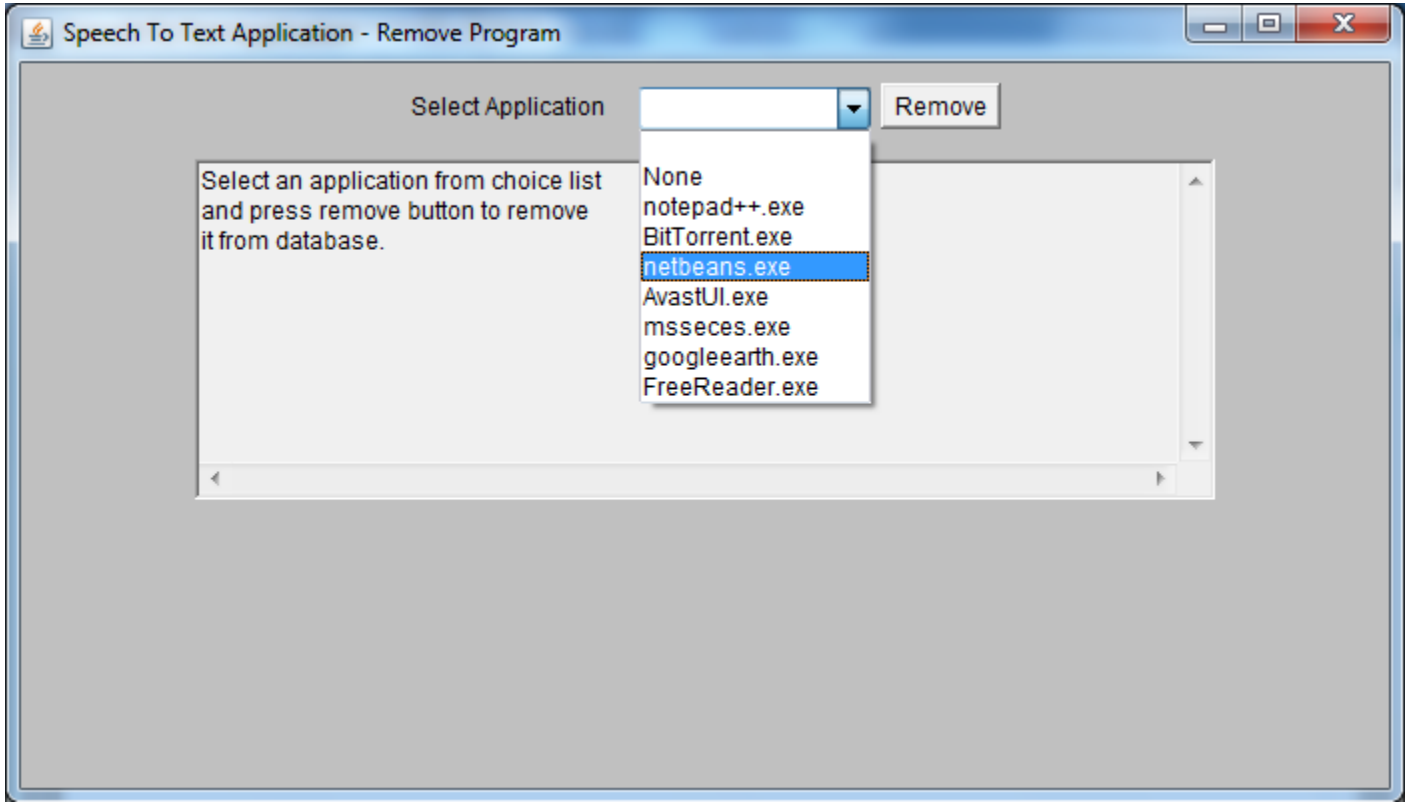
Basic idea

- Add Application : Here user have to specify absolute path to .exe file and alias that will be used for that application



Basic idea

- Remove Application : To remove any application user just have to select the particular application from list and press Remove button.



How it works?

- In the first when you run this program, it creates a JDBC:ODBC connection to our underlying database.
- It find outs different applications already added and the aliases for that apps accordingly.
- Then lists that applications in text area.

How it works?

- Whenever user presses the Speak to open button, first our application loads the particular grammar from a grammar file. Which looks something like below :

```
#JSGF 1.0 ISO8559-1;
grammar javax.speech.demo;

public <word> = note | good | hello | control | how | you | earth | fine | me;
```

- It also lists different Speech Engines and selects one.
- The speech engine can only recognize those words which are listed in this grammar file.

How it works?

- Now when user speaks anything the engine creates an event and accordingly engine tries to match the spoken word with entry in grammar file.
- Once system recognizes the spoken word which is given alias of any application, system again creates the database connection and finds the specific path for that application.
- And if that alias is proper than it will find an entry in database and that spoken application will start running!

Some testing results

No	Test Case	Expected Output	Avg. Accuracy (%)
1	Speaking a filename that exists in the grammar file.	Open particular application	60
2	Speaking a filename that is not in the grammar file.	Filename should not be recognized	90
3	User doesn't speak for a considerable time.	Don't do anything	90

Some testing results

Word Used	Test case 1	Test case 2	Test case 3
Hello	Executed	Not recognized	Automatically opens
How	Executed	Not recognized	Idle
Good	Not executed	Not recognized	Idle
Fine	Not executed	Not recognized	Idle
You	Executed	Not recognized	Idle
Me	Executed	Not recognized	Idle
Bittorrent	Not executed	Not recognized	Idle
Note	Executed	Recognized	Idle
Notepad	Executed	Not recognized	Idle
Avast	Not executed	Not recognized	Idle

Advantages

- Provides friendly user interface
- Reduces human efforts
- Saves time
- Helpful for physically disabled

Disadvantages

- As this Java Speech API Implementation is new it lacks of efficiency.
- Sometimes it recognizes same word more than one time.
- Supports single specified language.

Conclusions

- It provides easy way of computing
- Engine supports only limited words that are defined in grammar file.
- The efficiency of recognizing the spoken word is less and that is the main problem.

References

- Java speech API guide
- Cloudgarden documentation
- Wikipedia