Информационные технологии

УДК 004.522

В.С. СУДАКОВ, Э.Р. НОВИКОВ
(sudakovvs@yandex.ru, novikover2@gmail.com)
Российский экономический университет им. Г.В. Плеханова

ИСПОЛЬЗОВАНИЕ MICROSOFT SPEECH API ДЛЯ ОСУЩЕСТВЛЕНИЯ ЧЕЛОВЕКО-КОМПЬЮТЕРНОГО ВЗАИМОДЕЙСТВИЯ*

Целью статьи является демонстрация возможностей интерфейса Microsoft Speech API для работы с речью. Рассматриваются архитектура, преимущества и недостатки Microsoft Speech API, а также приводится пример программной реализации синтеза речи с использованием среды разработки Visual Studio.Net и языка программирования Visual Basic.Net.

Ключевые слова: *Speech API, распознавание речи, синтез речи, VisualStudio.Net, COM-технологии.*

На сегодняшний день в мире растет потребность в использовании технологий для распознавания и синтеза речи, которые позволяют сократить временные затраты пользователей на выполнение однотипных задач. Одним из таких инструментов является “Speech API”, основанный на технологии “COM”, разработчиком которого является компания “Microsoft”. Объект исследования: распознавание и синтез человеческой речи. Предмет исследования: применение “MS Speech API” для распознавания и синтеза речи.

Постановка задачи: провести анализ архитектуры “MS Speech API”, а также преимуществ и недостатков “MS Speech API”, продемонстрировать возможность использования “MS Speech API” для синтеза речи.

Методы решения задач: анализ документации “MS Speech API”, использование объектно-ориентированной парадигмы программирования для разработки программного приложения.

Решение задачи

Интерфейс программирования приложений для работы с речью (MS Speech API) в операционной системе “MS Windows” представляет собой инструмент, позволяющий реализовать один из вариантов человеко-компьютерного взаимодействия (голосовое взаимодействие) [1]. Этот API позволяет разработчикам создавать приложения, способные распознавать речь пользователя, а также осуществлять синтез речи, преобразуя текстовую информацию в аудио. Впервые “MS Speech API” (версия 1.0) начали использовать в 1990-х годах с появлением операционной системы “MS Windows 95”. Далее для операционных систем “Windows 98, 2000” появились версии “Speech API 2, 3, 4”. С появлением “Windows XP” была создана версия “Speech API 5.1”. Далее “Speech API” усовершенствовался за счет добавления распознавания новых языков и улучшения алгоритмов распознавания речи, и в конечном итоге для ОС “Windows 10 и 11” были разработаны версии “Speech API 5.3 и 5.4”.

Архитектура “MS Speech API” состоит из следующих частей [2, 4, 5]:

1. Компонент “Speech Recognition Engine” (SRE), отвечающий за распознавание речи. Он принимает звуковой сигнал с микрофона и преобразует его в информацию, представленную в виде текста. SRE способен работать с различными языками и диалектами.

2. Компонент “Speech Synthesis Engine” (SSE), отвечающий за синтез речи из текста. Он берет текст и преобразует его в аудиосигнал, который может быть воспроизведен через динамики или наушники.

3. Стандарт “Speech Recognition Grammar Specification” (SRGS), определяющий грамматику, которую SRE использует для распознавания. Пользователи могут создавать грамматики, которые определяют, какие команды или фразы приложение должно распознавать.

4. Стандарт “Speech Synthesis Mark up Language” (SSML), позволяющий управлять синтезом речи, обрабатывая интонацию, паузы и произношение.

* Работа выполнена под руководством Попова А.А., кандидата технических наук, доцента кафедры Информатики ФГБОУ ВО «РЭУ им. Г.В. Плеханова».

5. Инструменты и библиотеки, которые упрощают интеграцию “MS Speech API” и программных приложений и позволяют настраивать параметры, характеризующие синтез и распознавание речи.

Распознавание речи представляет собой сложный процесс, состоящий из нескольких шагов:

1. Захват звука. Интерфейс “Speech API” получает на вход аудиодорожку, записанную пользователем с помощью микрофона или иного аудиоисточника.

2. Преобразование звука в текстовый формат. Полученный аудиофайл обрабатывается, отбрасывается лишний фон и шум. После этого происходит распознавание полученных данных, используя акустические и лингвистические модели, встроенные в “Speech API”.

3. Обработка результата распознавания. После того как аудио было преобразовано в текст, технология распознавания дополнительно обрабатывает получившиеся данные, учитывая особенности языка, исправляя опечатки, разбивая текст на предложения, проставляя знаки препинания и другие элементы письменности.

4. Отправка результата. Данные, полученные с помощью “Speech API”, отправляются программному приложению, которое взаимодействует со “Speech API”.

Синтез речи из текста (TTS) можно представить в виде последовательности выполнения следующих шагов:

1. Обработка текста. Перед преобразованием в речь, текст необходимо профильтровать, разбив его на слова, фразы и предложения, учитывая контекст полученных данных.

2. Фонематический анализ. Перед созданием аудиосигнала, текст преобразовывается в фонемы, которые являются минимальной смыслоразличительной единицей любого языка.

3. Генерация речи. На этом этапе каждая из групп фонем анализируется и преобразуется в акустический сигнал. Процесс преобразования включает в себя определение частот и длительности для каждой фонемы, а также плавность перехода между ними.

4. Синтез речи. Созданные акустические сигналы объединяются, формируя единый аудиопоток, которые может быть передан в приложение в формате аудиодорожки.

Технология TTS в “Windows” использует разные алгоритмы для генерации речи. В основном для синтеза используется конкатенация звуков и фраз, а также синтез на основе глубокого обучения (DLS), где используются нейронные сети изучения зависимостей между текстом и звуком (RNN, CNN, GAN).

Среди преимуществ технологии “MS Speech API” можно выделить:

1. Наличие функций для распознавания и синтеза речи, позволяющих программным приложениям преобразовывать аудио в текст (распознавание речи) и текст в аудио (синтез речи).

2. Обеспечение помощи в создании коммуникаций для лиц с ограниченными возможностями здоровья.

3. Возможность управления приложениями и устройствами с помощью голосовых команд.

4. Способность интеграции с другими программными приложениями.

К недостаткам “MS Speech API” можно отнести:

1. Неточность в распознавании речи при работе со сложными акцентами и в условиях сильного фонового шума.

2. Поддержка ограниченного числа языков и диалектов.

3. Возможность передачи данных в “Microsoft”, что можешь нарушать требования конфиденциальности данных для компаний и отдельных пользователей.

4. Работает только в составе операционной системы “MS Windows”.

5. Дикторозависимость при распознавании речи.

Чтобы продемонстрировать возможность использования “MS Speech API” для синтеза речи, применим объектно-ориентированную парадигму программирования с использованием “VisualStudio.Net” и языка программирования “VisualBasic.Net”. Программное приложение будет преобразовывать введенный пользователем текст в речь. Для этого в среде разработки “VisualStudio.Net 2019” создается новый проект «Приложение Windows Forms (.NET Framework)» для “Visual Basic”[3]. Сначала необходимо присоединить к проекту объектную модель “MS Speech Object Library 5.4”. Для этого необходимо с помощью меню «Проект» зайти в выпадающее меню и выбрать строку «Добавить ссылку». После

этого в появившемся диалоговом окне «Менеджер ссылок» зайти на вкладку «COM» и выбрать “MS Speech Object Library 5.4”.

Далее с использованием панели инструментов необходимо настроить пользовательский интерфейс, состоящий из следующих элементов управления: один TextBox, один ComboBox, два TrackBar, три Label и три Button.

С помощью свойства «Text» в окне «Свойства» необходимо изменить название формы на «Speech API». Для элементов управления Button1, Button2 и Button3 с помощью свойства Name в окне свойств необходимо набрать, соответственно, имена BtnSpeak, BtnPause и BtnResume. С помощью свойства Textв окне свойств необходимо задать элементам управления Button с именами BtnSpeak, BtnPause, BtnResume, соответственно, названия «Произнести», «Пауза» и «Возобновить».

Для элемента управления TextBox1 с помощью свойства Name в окне свойств необходимо набрать YourText. Свойству «Multiline» элемента управления TextBox1 в окне свойств присвоить значение «True» для того, чтобы в поле элемента управления можно было отображать более одной строки текста. Для этого же элемента в окне свойств добавить вертикальную полосу прокрутки для того, чтобы при загрузке больших текстов можно было увидеть ту часть, которая не попала в видимую область элемента управления.

Элементы управления ComboBox1 и TrackBar1, TrackBar2 необходимо разместить на форме снизу от элемента управления TextBox1. Для элемента управления ComboBox1 с помощью свойства Name в окне свойств необходимо набрать ComboBoxOpenVoice. Элемент управления ComboBox1 будет выводить список всех доступных голосов, с помощью которых можно произнести текст. Для элементов управления TrackBar1 и TrackBar2 с помощью свойства Name в окне свойств необходимо набрать, соответственно, Track Bar Rate и Track Bar Volume.

Элементы TrackBar1 и TrackBar2 с именами TrackBarRate и TrackBarVolume отвечают за регулировку громкости и скорости воспроизведения текста. С помощью свойств «Minimum» и «Maximum» в окне свойств определяется диапазон значений, которые они могут принимать. Для ползунка, отвечающего за громкость, минимум будет 0, а максимум 100. Для ползунка, отвечающего за скорость, минимум будет минус 10, а максимум 10.

Слева от элемента управления ComboBox1 разместить элемент управления Label1. Для элемента управления Label1 с помощью свойства «Text» в окне свойств необходимо набрать текст «Выбор голоса». Поместить оставшиеся элементы управления Label2 и Label3 слева от элементов управления TrackBar1 и TrackBar2 и назвать их соответственно «Громкость» и «Скорость» с помощью свойства «Text» в окне свойств. Для того, чтобы можно было изменять размеры элементов управления Label, необходимо их свойству «AutoSize» присвоить значение False.

Цвет фона, размер и параметры шрифта, расположение текста внутри элементов управления настраиваются с помощью свойств «BackColor», «Font», «TextAlign», «ForeColor» в окне свойств.

В результате пользовательский интерфейс программного приложения будет выглядеть следующим образом (см. рис. 1.).

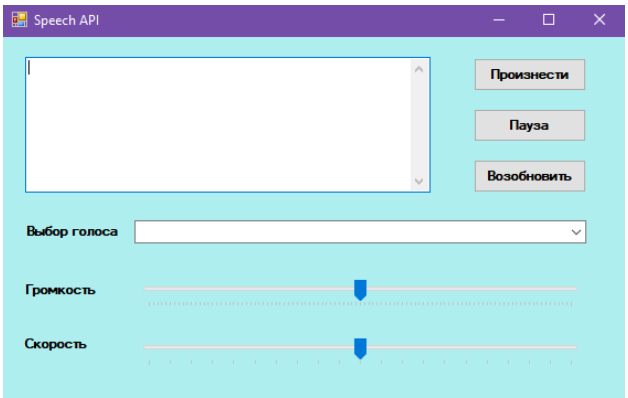

Рис. 1. Пользовательский интерфейс программного приложения

При разработке программного кода сначала необходимо подключить пространство имен SpeechLib для работы с речью (верхняя часть программного кода на рис. 2).

Далее необходимо объявить переменную SAPI – объект, который будет использоваться для работы с голосовым синтезом. Создать экземпляр класса SpVoice, который связывается с объектом SAPI. В обработчике события Form1_Load (загрузка формы) с помощью свойств «Value» для элементов управления TrackBar1 и TrackBar2 выставить исходные положения ползунков «Громкость» и «Скорость». Для скорости воспроизведения начальным значением будет 0 (обычная скорость без ускорений и замедлений), показатель громкости по умолчанию будет на уровне 50.

```
Imports SpeechLib

Ссылок: 2
Public Class Form1
    Dim SAPI As Object = CreateObject("SAPI.spvoice")
    Private WithEvents Voice As SpVoice = SAPI
    Ссылок: 0
    Private Sub Form1_Load(sender As Object, e As EventArgs) _
        Handles MyBase.Load
        TrackBarRate.Value = 0
        TrackBarVolume.Value = 50
        For i = 0 To SAPI.getvoices.count
            Try
                SAPI.Voice = SAPI.getvoices.item(i)
                ComboBoxOpenVoice.Items.Add("Голос #" & i _
                    & " " & SAPI.Voice.GetDescription)
            Catch
            End Try
        Next
    End Sub
```

Рис. 2. Программный код программного приложения

С помощью цикла For производится определение всех доступных голосов, имеющихся в системе, и добавление их в выпадающий список ComboBoxOpenVoice (элемент управления ComboBox1). Программный код для кнопки с именем «Произнести» приведен на рис. 3.

```
Ссылок: 0
Private Sub BtnSpeak_Click(sender As Object, e As EventArgs) _
    Handles BtnSpeak.Click
    If ComboBoxOpenVoice.SelectedIndex = -1 Then
        MessageBox.Show("Пожалуйста, выберите голос перед произношением текста")
        Return
    End If
    SAPI.Speak(YourText.Text, AudioPlayMode.Background)
End Sub
```

Рис. 3. Программный код элемента управления Button с названием «Произнести»

В программном коде на рис. 4 на с. 11 сначала производится проверка выбора голоса из списка, выпадающего в элементе управления ComboBox. Проверка производится с помощью условного оператора If в обработчике события BtnSpeak_Click. Индекс равный ‘-1’ означает, что пользователь не выбрал голос в элементе управления ComboBox1, после чего он получит предупреждающее сообщение о необходимости выбрать голос. Если голос выбран, то производится чтение текста из элемента управления TextBox с именем YourText с помощью вызова метода SAPI.Speak, который отвечает за старт голосового воспроизведения текста. С помощью параметра AudioPlayMode.Background реализована возможность изменения громкости и/или скорости прямо во время произношения текста, без прерывания или окончания его воспроизведения. Воспроизведение текста может быть приостановлено и возобновлено с помощью элементов управления Button2 и Button3 с названиями «Пауза» и «Возобновить» соответственно. Скорость и громкость голосового воспроизведения текста меняется за счет перемещения ползунков на элементах управления TrackBar1 и TrackBar2. Программный код для указанных выше элементов управления выглядит следующим образом (см. рис. 4 на с. 11).

```
Ссылок: 0
Private Sub TrackBarRate_Scroll(sender As Object, e As EventArgs) _
    Handles TrackBarRate.Scroll
    SAPI.rate = TrackBarRate.Value
End Sub

Ссылок: 0
Private Sub TrackBarVolume_Scroll(sender As Object, e As EventArgs) _
    Handles TrackBarVolume.Scroll
    SAPI.volume = TrackBarVolume.Value
End Sub

Ссылок: 0
Private Sub BtnPause_Click(sender As Object, e As EventArgs) _
    Handles BtnPause.Click
    SAPI.pause
End Sub

Ссылок: 0
Private Sub BtnResume_Click(sender As Object, e As EventArgs) _
    Handles BtnResume.Click
    SAPI.resume
End Sub
```

Рис. 4. Программный код элементов управления TrackBar и Button (с именами«Пауза» и «Возобновить»)

В результате выбора голоса в выпадающем списке, реализуемом с помощью элемента управления ComboBox1, программное приложение получает индекс выбранного голоса и, на основе этого индекса, устанавливает голос с помощью свойства объекта SAPI. Программный код для элемента управления ComboBox1 приведен на рис. 5.

Также с помощью обработчика событий SAPI_Highlight в программном приложении реализовано выделение активных слов, т. е. каждое слово при его произношении подсвечивается в элементе управления TextBox. Программный код для элемента управления ComboBox1 приведен на рис. 6.

```
Ссылок: 0
Private Sub ComboBoxOpenVoice_SelectedIndexChanged(sender As Object, e As EventArgs) _
    Handles ComboBoxOpenVoice.SelectedIndexChanged
    Dim selectedIndex As Integer = ComboBoxOpenVoice.SelectedIndex
    If selectedIndex >= 0 AndAlso selectedIndex < SAPI.GetVoices().Count Then
        SAPI.Voice = SAPI.GetVoices().Item(selectedIndex)
    End If
End Sub
```

Рис. 5. Программный код элемента управления ComboBox1

```
Ссылок: 0
Private Sub SAPI_Highlight(SreamNumber As Integer, StreamPosition As Object,
                           CharacterPosition As Integer, Length As Integer) _
                           Handles Voice.Word
    YourText.Select(CharacterPosition, Length)
    YourText.Focus()
End Sub
```

Рис. 6. Реализация выделения произносимых слов

Для демонстрации синтеза речи необходимо запустить программное приложение. В пользовательском интерфейсе пользователю будет предлагаться список доступных голосов (см. рис. 7 на с. 12).

Далее необходимо набрать текст на русском или английском языке в элементе управления TextBox1. Из выпадающего списка с предлагаемыми голосами (в элементе управления ComboBox1) необходимо выбрать голос для чтения (русский или английский голос, в зависимости от языка, на котором набран текст). В случае, если пользователь не выберет ни один из голосов, он получит пре-

дупреждающее сообщение о том, что необходимо выбрать голос. До тех пор, пока голос не будет выбран, программное приложение не продолжит работу. После выбора голоса необходимо нажать на кнопку «Произнести». Приложение начнет воспроизводить текст и выделять произносимые слова. С помощью перемещения бегунков на элементах управления TrackBar1 и TrackBar2 можно произвести настройки громкости и скорости воспроизведения речи. С помощью кнопки с именем «Пауза» можно поставить процесс чтения текста на паузу, а с помощью кнопки «Возобновить» – возобновитьчтение текста (см. рис. 8).

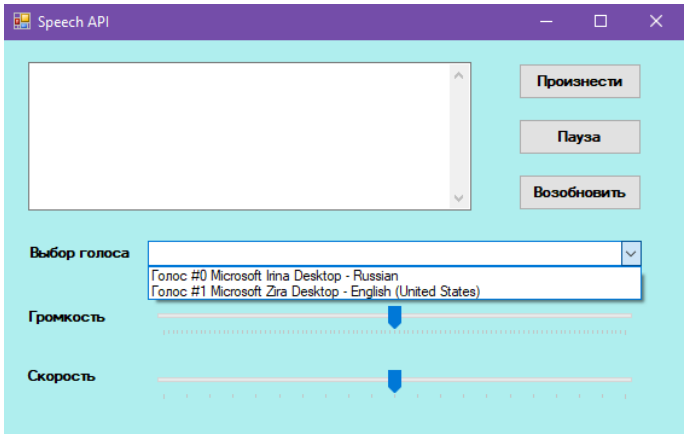

Рис. 7. Выбор доступных голосов

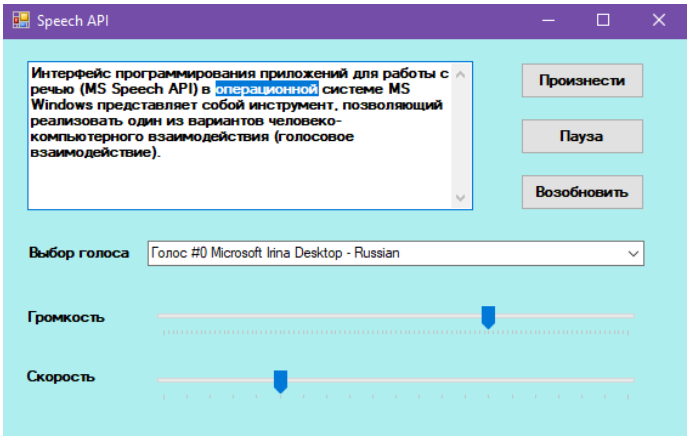

Рис. 8. Демонстрация работы программного приложения

Использование программного приложения показывает, что программная реализация чтения текстов с использованием MS Speech API не требует разработки сложного программного кода. Поэтому такое приложение может быть интегрировано со сторонними программными приложениями. При этом, для осуществления интеграции должно быть дополнительно разработано программное приложение для передачи текста, подлежащего чтению, из сторонних программных приложений.

Таким образом, в данной статье проанализирована архитектура MS Speech API, рассмотрены его достоинства и недостатки, а также приведен пример реализации программного приложения, преобразовывающего текст в речь. Программная реализация чтения текстов с использованием MS Speech API не требует написания сложного программного кода. Поэтому разработанное приложение может быть интегрировано с другими программными приложениями.

Литература

1. Попов А.А., Батраков В.А., Вербицкий А.С. Человеко-машинное взаимодействие. М.: Изд-во Военной академии Ракетных войск стратегического назначения им. Петра Великого, 2015.
2. Речь, голос и беседа в Windows 11 и Windows 10 // Microsoft. [Электронный ресурс]. URL: https://learn.microsoft.com/ru-ru/windows/apps/develop/speech (дата обращения: 18.10.2023).
3. Учебник. Создание приложения WinForms на VisualBasic // Microsoft. [Электронный ресурс] URL: https://learn.microsoft.com/ru-ru/visualstudio/ide/create-a-visual-basic-winform-in-visual-studio?view=vs-2022 (дата обращения: 19.10.2023).
4. Чесебиев И.А. Компьютерное распознавание и порождение речи. М.: Спорт и Культура, 2008.
5. Microsoft Speech API (SAPI) 5.3 // Microsoft. [Электронный ресурс]. URL: https://learn.microsoft.com/en-us/previous-versions/windows/desktop/ms723627(v=vs.85) (дата обращения: 19.10.2023).

VLADISLAV SUDAKOV, EDUARD NOVIKOV
Plekhanov Russian University of Economics

THE USE OF MICROSOFT SPEECH API FOR HUMAN-COMPUTER INTERACTION

The purpose of the paper is to demonstrate the potential of the interface of Microsoft Speech API for the work with speech. There are considered the architecture, advantages and disadvantages of Microsoft Speech API. There is given the example of the software implementation of the speech synthesis with the use of the development environment "Visual Studio.Net" and the programming language "Visual Basic.Net".

Key words: *Speech API, speech recognition, speech synthesis, VisualStudio.Net, COM-technologies.*