**Олжас Аймухамбетов**

**(Астана, Казахстан)**

**Использование возможностей SIML**

**для создания чат-бота на C#**

**Введение**

Чат-бот – программа, имитирующая человеческое общение. Хотя первые реализации чат-ботов возникли ещё в ХХ веке, в настоящее время наблюдается стремительный рост популярности этого направления в связи с тем, что данная технология позволяет автоматизировать многие процессы максимально понятно для пользователя любого уровня.

В данной статье приводится пример использования языка разметки SIML при разработке чат-бота на базе C#.

Требования к разработке: Syn Chatbot Studio, MS Visual Studio, .NET Framework 4.5 и выше. При составлении данной статьи использовалась среда разработки MS Visual Studio 2013.

**Настройка графического пользовательского интерфейса**

При разработке чат-бота следует учитывать, что программная часть, выполненная на С# является лишь графическим отображением. Ядром чат-бота, содержащим базу знаний (вопросы и ответы) и логику ведения диалога, является SIML.

Для начала в среде разработки Visual Studio необходимо создать приложение WPF.

Для взаимодействия с нашим Chatbot нам понадобятся следующие компоненты, которые будут добавлены в наше приложение WPF:

* Поле ввода для отправки введенного сообщения;
* Кнопка «Отправить» для отправки текстового сообщения;
* Окно вывода, чтобы отобразить ответ, сгенерированный чат-ботом.

Для наличия всех вышеуказанных пунктов необходимо вставить в файл *MainWindow.xaml* следующий код:

<Window x:Class="SIML\_Chatbot\_Demo.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Title="SIML Chatbot Demo" Height="800" Width="600">

<Grid>

<Grid.RowDefinitions>

<RowDefinition Height="250\*"/>

<RowDefinition Height="5\*"/>

<RowDefinition Height="50\*"/>

</Grid.RowDefinitions>

<TextBox Name="OutputBox" IsReadOnly="True" />

<DockPanel Grid.Row="2">

<TextBox Width="500" Name="InputBox" />

<Button Content="Send" Name="SendButton" Click="SendButton\_OnClick"/>

</DockPanel>

</Grid>

</Window>

Далее необходимо добавить обработчик событий для Clik= “SendButton\_OnClick”:

private void SendButton\_OnClick(object sender, RoutedEventArgs e)

{

//Our code will go here

}

**Импорт пакета Syn.Bot NuGet**

Чтобы импортировать переносимую библиотеку классов из NuGet, **Tools->NuGet Package Manager->Package Manager Console** и введите Install-Package Syn.Bot. Библиотека классов будет добавлена в проект и на нее будут ссылаться должным образом. Syn.Bot является официальным переводчиком для SIML и регулярно обновляется.

**Использование IDE для SIML**

Для создания базы знаний необходимо создать пустой шаблон в Syn Chatbot Studio посредством: **File–>New–>Project**.

**Основы SIML**

Для создания базы знаний используется разметка SIML – *Synthetic Intelligence Markup Language – «Язык разметки синтетического интеллекта»*.

Стандартная «модель» (связка «вопрос»-«ответ»), написанная на SIML, выглядит следующим образом:

<SIML>

<Concept Type=”public” Name=”Hello world”>

<Model>

<Pattern>Привет, бот!</Pattern>

<Response>Привет, человек</Response>

</Model>

</Concept>

</SIML>

Где:

SIML – открывающий и закрывающий тег файла.

Concept (***Концепция***) – используется для группирования Моделей по тематике.

Model (***Модель***) – базовая единица знаний в SIML, содержит вопрос и ответ. Каждая модель может относится только к одной Концепции. Разрешается использование нескольких Шаблонов в рамках одной модели.

Pattern (***Шаблон***) – набор токенов/слов, которые проверяются во время выполнения по отношению к пользовательскому вводу, чтобы проверить, соответствует ли пользовательский ввод указанной последовательности токенов/слов.

Response (***Ответ***) – в случае совпадения запроса, введённого пользователем, с одним из шаблонов, пользователю возвращается строка/данные, содержащиеся в теге Ответ, который относится к той модели, что и Шаблон.

Следует отметить, что возможности SIML гораздо шире стандартных строковых Шаблонов и Ответов – пользователю предоставляется возможность использовать регулярные выражения, вставлять JavaScript, использовать операторы поиска, ключевые слова, переменные и т.д. Все возможности с примерами описаны на сайте разработчика Syn Developer Network.

**Экспорт базы знаний в файл и использование в WPF приложении**

После того, как создана база знаний, необходимо выгрузить её в отдельный файл. В среде разработки Syn Chatbot Studio при открытом проекте необходимо выбрать **Project–>Export to Package**. После чего указать название файла и путь к катологу *Bin* в папке, содержащей созданный проект C#.

Для того, чтобы WPF приложение могло использовать экспортированную базу знаний, необходимо вставить следующий код в файл MainWindow.xaml.cs:

using System.IO;

using System.Windows;

using Syn.Bot;

namespace SIML\_Chatbot\_Demo

{

public partial class MainWindow

{

public SynBot Chatbot;

public MainWindow()

{

InitializeComponent();

Chatbot = new SynBot();

Chatbot.PackageManager.LoadFromString(File.ReadAllText("primer.simlpk" ));

}

private void SendButton\_OnClick( object sender, RoutedEventArgs e)

{

var result = Chatbot.Chat(InputBox.Text);

OutputBox.Text = string .Format("User: {0}\nBot: {1}\n{2}", InputBox.Text, result.BotMessage, OutputBox.Text);

InputBox.Text = string .Empty;

}

}

}

После запуска WPF приложения Пользователю будет доступно окно со строкой ввода и окном вывода. В зависимости от ввода пользователя в окне отображения будет отображена переписка пользователя с чат-ботом на основании используемой базы знаний (в коде – файл с названием “primer.simlpk”, результат экспорта базы знаний из студии в файл).

**Литература:**

1. База знаний по работе с продуктами компании Synthetic Intelligence Network – <https://developer.syn.co.in/tutorial/index.html>
2. Мэтьюс Д. – Add a Chatbot to a C# Application using SIML (Synthetic Intelligence Markup Language) – <https://www.codeproject.com/Articles/866760/%2FArticles%2F866760%2FAdd-a-Chatbot-to-a-Csharp-Application-using-SIML-S>, 2017

**Научный руководитель:**

Ла Лира Львовна