Advanced Computer Lab Report

IVR Train Reservation System

Hossam Saraya 13-7985
Tarek Sheasha 16-4127

January 2013

Contents

1 Introduction

1.1 About the project

In a nutshell, this paper explains the basic concepts VoiceXML. We created a simple IVR train reservation application to demonstrate how VoiceXML can be used along with Voxeo to deliver a telephony application that is based on keyboard input and/or voice recognition. The application workflow will be explained in more detail as you read the paper.

1.2 Technologies used

1.2.1 VoiceXML

VoiceXML is the W3C's standard XML format for specifying interactive voice dialogues between a human and a computer. It allows voice applications to be developed and deployed in an analogous way to HTML for visual applications. Just as HTML documents are interpreted by a visual web browser, VoiceXML documents are interpreted by a voice browser. A common architecture is to deploy banks of voice browsers attached to the Public Switched Telephone Network (PSTN) to allow users to interact with voice applications over the telephone.[1]

1.2.2 Voxeo

Voxeo Corporation is a technology company that specializes in providing development platforms for unified customer experience (self-service) and unified communications (real time communications) applications. Voxeo's products are all available as SaaS / Platform as a service (public cloud), on-premise software (private cloud), or hybrid clouds that combine the public-cloud and on-premise components.[2]

2 How to use?

Our application is meant to be a telephony application. There are three options for using the application

Skype VoIP Simply by calling **+990009369996171339** from Skype.

Voxeo SIP Through URL **sip:9996171339@sip.voxeo.net**

Hosted SIP Through URL **sip:vxml@127.0.0.1**

After following any of the previous option. The user is presented with a welcome message asking him whether he wants to use the keyboard or his voice to use the service. In both cases, the user is asked about the desired departure place, destination and the date. The options for the user choices are mentioned below.

Departure The following values are valid

- Cairo
- Alex
- Aswan
- Luxor

Destination The following values are valid

- Cairo
- Alex
- Aswan
- Luxor

Day Values from 1 to 31 are valid

Hour Values from 1 to 24 are valid

3 What happens under the hood

Voxeo maps the url of our vxml file with a SIP phone number so that when that number gets called, Voxeo requests our vxml url and it starts executing the vxml code. It's that simple.

4 Implementation

Everything regarding the implementation is located in the main vxml file. The following is the explanation of some important snippets in the implementation.

4.1 Greeting the user

In the very beginning of the session. The user is greeted with a welcome message asking him to choose how the session will continue (voice or non-voice).

```
<menu id="menu">
  <property name="inputmodes" value="dtmf"/>
  <prompt bargein="false">
      Welcome to our train station reservation system
    </prompt>
  <prompt>
    <enumerate>
      <value expr="_prompt"/> press <value expr="_dtmf"/>
    </enumerate>
  </prompt>
  <choice next="#dtmfForm_N_From" dtmf="1">
      for non voice reservation
    </choice>
  <choice next="#dtmfForm_V_From"  dtmf="2">
      for voice reservation
    </choice>
</menu>
```

Based on the user input, he will be redirected to different form elements. `#dtmfForm_N_Form` is the non-voice form, and `#dtfForm_V_Form` is the voice based form. The choice of the user is accessed through the dtmf attribute. From now on, we will be explaining the voice related snippets because that's what we are concerned about in this course.

4.2 Form Fields

In our application, we ask the user for input. If the application was desktop or web based, it would have UI components to let the user submit his input to the application. It is not much different in our application, the `<field>` element is responsible for holding the user input after recognizing the user input using the best match approach. The options for the user are nested in a ¡one-of¿ element: Think of it as a combo box. The user is asked to enter to say the departure place. This can be observed in the following snippet.

```
<field name="fromField" slot="fromField">
  <grammar mode="voice" root="from">
    <rule id="from" scope="public">
      <one-of>
        <item>Cairo</item>
        <item>Alex</item>
        <item>Luxor</item>
      </one-of>
    </rule>
  </grammar>
  <prompt bargein="false">
  Please say your departure place, from the comeing places
  <prompt>
Cairo,
Alex,
Luxor,
Aswan.
</prompt></prompt>
</field>
```

After a grammar match has occured, we do some checks and validations to validate the user input. This is achieved by using the `<filled>` element.

```
<filled namelist="destinationField" mode="all">
  <if cond="fromField == destinationField">
    <prompt bargein="false">
  Sorry, departure place and destination are the same try again.
</prompt>
    <clear namelist="destinationField"/>
    <else/>
    <prompt>
Looks like you said <value expr="destinationField" />.
</prompt>
  </if>
</filled>
```

4.3 The Output

After receiving all the needed input from the user. The price of the ticket is calculated using the very simple javascript function we defined.

```
<script>
function price(Number,Class)
{
var res=0;
if(Number==’one’)
res+=10;
if(Number==’two’)
res+=20;
if(Number==’three’)
res+=30;
if(Number==’four’)
res+=40;
if(Number==’five’)
res+=50;
if(Number==’six’)
res+=60;
if(Number==’seven’)
res+=70;
if(Number==’eight’)
res+=80;
if(Number==’nine’)
res+=90;
if(Class==’third’)
res+=50;
if(Class==’second’)
res+=150;
if(Class==’first’)
res+=200;
return res;
}
</script>
```

In addition to that, after each user input, the user is asked to confirm his choice.

5 Future work

The project was very interesting yet we didn't care for any aspects of the project except for the voice recognition part because that's what we wanted to demonstrate. The project is incomplete as the price function is just a dummy function for price calculations and it can not be used in a real system. In addition to that, The departure, arrival and date captured from the user input are not used at all due to the fact that the project is just for demonstration and it can not be applied in real life in that state.

References

[1] http://en.wikipedia.org/wiki/VoiceXML.

[2] http://en.wikipedia.org/wiki/Voxeo.