

Dialog planning in VoiceXML

Csapó Tamás Gábor <csapot AT tmit.bme.hu>
Zainkó Csaba

4 January 2011

4. Measuring tasks

Throughout the measurement, we will create some sample VoiceXML dialogs. The code can be written as plain text (e.g. using the Notepad++ program [1]). For the initial syntax checking and debugging the BladeWare development environment [2] will be used. The completed programs will be uploaded to a web server; will be tried live calling the sample application using X-Lite [3]. You can find a detailed description for each VoiceXML label in [4].

During solving the task, you must strive to create a system that real users could comfortably use (e.g. short, understandable and transparent menu prompts are encouraged during application development).

[1] <http://notepad-plus.sourceforge.net/hu/site.htm>

[2] <http://www.commetrex.com/products/CTMiddleware/bladewareVXML.html>

[3] <http://www.counterpath.com/x-lite-download.html>

[4] <http://www.vxml.org/>

4.1 Main menu

Create a syntactically correct VoiceXML program (`index.vxml`), which welcomes the user! The skeleton of the solution:

```
<form>
  <block>
    Welcome.
  </ block>
</form>
```

Test your program in the BladeWare environment. Place `index.vxml` file in the root directory of the measurement, and then start the program with the `runbladeware.bat`. Here, we only have a console available, which prints what the real system would say, and you type what you would respond to this. Examine the output generated by the program, and attach it to your report!

If the code is executed successfully, upload the file to the web server into the directory of your measurement group and try the application using X-Lite!

The upload page can be found at <http://hermes.tmit.bme.hu/meresX/upload.html> which places the file into meresX directory. Instead of "X" use the number corresponding to your measurement group! Call 1234 using the X-Lite software, where you can reach the main VoiceXML page. After you listen to the announcement, speak / press the appropriate number and you will be redirected to the `index.vxml` of that group.

Hereafter, before uploading your VoiceXML code to the web server, always check in BladeWare if it works without error.

Complete the existing code to a 3-point menu within which you can navigate:

```
-- Main menu
|-- Menu1
|-- Menu2
|-- Menu3
|-- Exit
```

For this task creating a menu is needed, in which you can navigate with speech and DTMF codes. In VoiceXML the individual elements may be assigned by an identifier ("id" attribute), to which you can refer later. With pressing the keys on the phone you can send DTMF codes, which is also a way of navigation through the menu.

The skeleton of the solution:

```
<menu id="mainmenu" dtmf="true">
  <prompt>
    Welcome.
    <enumerate/>
  </prompt>
  <choice next="#menu1" dtmf="1">
    Menu1
  </choice>
  ...
</menu>
<form id="menu1">
  ...
</form>
```

You can create a simple menu with the `<menu>` tag, which presents some options to choose from. The options are listed with the `<choice>` tag, and include the "next" in their properties, where to go further. This may be another element within the same document starting with # (of which we can refer to the "id" property), other file in that directory, or an arbitrary reference (e.g., another webpage).

After the Menu2 and Exit menu items saying goodbye and exit should come (`<form>`, `<block>`, `<disconnect>`), after the Menu3 navigate back to the main menu (`<form>`, `<block>`, `<goto next="#mainmenu"/>`)!

Create the error handler instructions (`<nomatch>`, and `<noinput>` `<help>`)!

Test your system first in BladeWare and after that in X-Lite, record the audio with the REC button, and then attach it to your report!

4.2 Date

Create a form that will read out loud the current date! If a user enters the site he should hear the exact date / time! (Hint: `<var name="years" expr="new Date().getFullYear()"/>`; similarly `getMonth() +1`, `getDate()`, `getHours()`, `getMinutes()`).

Paste the new menu item into the main page, the modify `Menu1` to `Date`!

Alter the main message of the main page so that it welcomes the user depending on the time of day! This menu before `<form>` `<block>` and instructions to insert, then if-elseif-else to set the proper greeting. Finally, we need to go through `<goto>` instructions to the main menu, otherwise the call is disconnected.

Test this date function again in BladeWare and X-Lite as well.

4.3 Calculator

Now we prepare a speech-driven calculator introduced in the VoiceXML programming guide. For this, a speech recognition grammar is needed. This system does not include built-in grammar to recognize numbers / digits, so it should be explicitly described. Make a VoiceXML grammar suitable to recognize 0-9 digits (`<grammar>`, `<rule>`, `<one-of>`, `<item>`)!

Write a calculator that first asks for an operation (addition / subtraction / multiplication / division) and two numbers (0-9), then perform the desired action, and read out loud the results!

Paste the new menu item into the main page instead of `Menu2` (call it `Calculator`)!

After the calculation is made the program should ask to user if he wants to perform a new operation, or return to the main menu!

Include the factorial calculation as well! The program should ask only one number between 0-9, and read out loud the factorial of the number!

Hint: it is possible to include JavaScript source code within VoiceXML, so you can simply write a proper function:

```
<script>
  <![CDATA[ function factorial(n) {return (n <= 1) ? 1 : n *
    factorial(n-1);} ]]>
</script>
...
<value expr="factorial(n)"/>
```

4.4 Movie guide

In the next task you will write a program to query movie guide. Calling the system it will be possible to find films and cinemas. The data is in a pre-built SQL database, which you can retrieve using PHP.

Create a PHP page based on the provided sample files that connects to the SQL database, performs the query and disconnects. The skeleton of the solution:

```
"movie.php"
<?php
echo "<?xml version = \" 1.0 \"encoding = \"UTF-8 \"?> \n";
echo "<vxml version=\"2.0\" xmlns=\"http://www.w3.org/2001/vxml\"> \n";
```

```

// Database handling and
// Functions for generating VoiceXML grammars
require ".. / movie_function.php";

// generates a list of film titles
// return value e.g. "Agora, Spongebob"
movietitle_prompt();

// generates a grammar of film titles
// return value e.g. "<item> Agora </item>"
// <item> Spongebob </item> "
movietitle_grammar();

// generates a list of movie names
// return value e.g. "Allé, Tabán"
cinemaname_prompt();

// generates a grammar of movie names
// return value e.g. "<item> Allé </item>"
// <item> Tabán </item> "
cinemaname_grammar();

// Finds that in which day the given film
// is played in the given cinema
// return value e.g. "Monday, Wednesday"
// or: "The movie does not play."
findtime_prompt(string cinemaname, string movietitle);

```

Ask the chosen movie and film from the user in a VoiceXML dialog, and store it in e.g. the `movietitle` and `cinemaname` VoiceXML variables. Pass the result with the `<submit next="movie.php" method="post" namelist="movietitle cinemaname"/>` tag to a PHP page. In the PHP processing page, you can extract it (`$ _GET`);. Then perform a query, which results when the film is played at that cinema.

Paste the new menu item into the main page instead of Menu3! Test the function in BladeWare and record a sample of the dialogue with the REC button in X-Lite!

4.5 Rock-Paper-Scissors game (supplementary task)

Write the VoiceXML version of the Rock-Paper-Scissors game!

Sample: <https://studio.tellme.com/library2/code/ex-111/>