Interactive VoiceXML Module into SIP-Based Warning Distribution System

Karel Tomala, Jan Rozhon, Filip Rezac, Jiri Vychodil,
Miroslav Voznak, and Jaroslav Zdralek

VSB - Technical University of Ostrava, 17. listopadu 15,
70800 Ostrava, Czech Republic
{karel.tomala,jan.rozhon,filip.rezac,jiri.vychodil,
miroslav.voznak,jaroslav.zdralek}@vsb.cz

Abstract. This article discusses the use of the Voice Extensible Markup Language (VoiceXML, VXML) to create a complex voice menu in danger alert communication system. The system was created as a part of research at Department of Telecommunications at the VSB – Technical University of Ostrava. Creating a voice menu provides end-users more information about the impending danger as well as instructions on how to behave in a given situation. If users receive a pre-recorded warning message in the form of a phone call, it will provide a telephone number on which they can obtain more information. In order to achieve the desired functionality, we had to use open-source PBX Asterisk, the VoiceGlue package which features both the VoiceXML interpreter and the Text-to-Speech (TTS) module.

Keywords: Voice Extensible Markup Language, VoiceGlue, Text-To-Speech, Hypertext Preprocessor.

1 Introduction

Today, more than ever before, information plays an important role, in particular information received at the right moment. In order for a person to be able to make the right choice, s/he should have access to information when it is most needed. People are increasingly aware of this, as today information can be accessed not only through a computer or laptop but also through various PDAs, mobile phones and other communications devices. However, a situation may arise in which the only link to the outside world and the only way to access information will be the voice communication via a mobile phone. Speech is a fundamental and most important means to transfer information between human beings, so naturally there is an effort to use it in modern automated systems that surround us in everyday life.

Our interactive voice menu uses the VoiceXML markup language combined with some open-source tools. It was created as an extension on our own system to alert the public to danger by means of sending voice messages over the communication system directly to their terminal device (mobile phone, fixed line, etc.) [1]. Besides the warning, the delivered voice message also includes a phone number to dial to reach

A. Dziech and A. Czyżewski (Eds.): MCSS 2011, CCIS 149, pp. 338–344, 2011.

the voice menu which is located directly in the system. This is where the end-user obtains more information about the emergency as well as more instructions on how to address the problem occurred.

2 Used Technology

The comprehensive voice menu was designed using open-source software components Linux, VoIP Asterisk PBX [2], VoiceGlue (VXML language interpreter OpenVXI) [3] and Cepstral Swift text-to-speech module.

VoiceXML [4] enables to easily create complex voice menus that are considered an equivalent of web pages in the area of voice communications. VXML, as the name suggests, is a subset of Extensible Markup Language (XML) [5] and uses a precisely specified set of tags. Through these tags, the type and nature of voice conversation between the end-user and the voice automat can be unambiguously determined. The source code in the VXML format is processed by the VXML interpreter language to a user-friendly voice conversation format. Subsequently, the interpreter has to options to generate the voice message for transmission: to play pre-recorded messages in a given sequence or to feed the text input into the module processing text to speech (Text-To-Speech). The Text-to-Speech module is one of many VXML complements. Another example is a module that enables processing speech to text (Speech-To-Text). The VXML interpreter is perfectly capable of interpreting the VXML language. However, where necessary, advanced features such as Text-To-Speech module should be implemented, in particular if it is not clear in advance whether the system will need to interpret sound.

2.1 VoiceXML Document Structure

A VXML document consists of blocks linked by pairs of tags. In addition, it also contains one stand-alone VXML document in which all other blocks are nested. Figure 1 provides an overview of the fixed VXML document structure. Below the element level, any of the above listed types of blocks, such as a ‘prompt’ block, can be added.

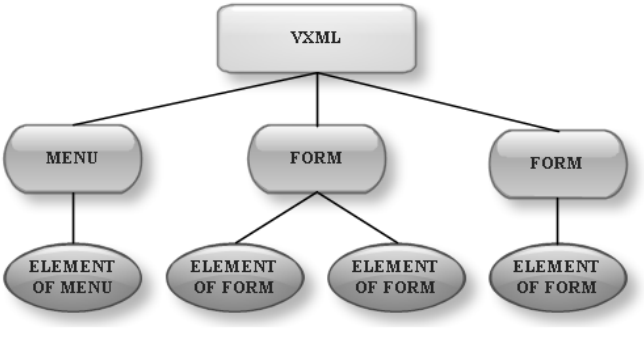

Fig. 1. Fixed structure of VXML document

3 Implementing VXML into Asterisk PBX

To be able to launch the interactive voice menu, PBX Asterisk needs to be configured. At the end-user's request, information about the emergency is read in the voice menu. In other words, a Hypertext Preprocessor (PHP) script that enables translating information entered into the web form into the VoiceXML and play it as a voice message had to be created.

The VoiceGlue package was applied to achieve the required functionality. It features both the VXML language interpreter and the TTS module. Voice output from the integrated TTS module was not sufficiently intelligible, and a separate TTS module had to be installed. The platform consists of the following open-source software components:

- OS Ubuntu x64 10.10,
- Asterisk PBX 1.6.2.15,
- VoiceGlue 0.12,
- Cepstral Swift.

The VoiceXML interpreter OpenVXI, Text-To-Speech module Cepstral Swift and the developed PHP script [6] to process information from a Web server to VXML were implemented directly into the Softswitch Asterisk [7]. This script is described in Chapter 4. The platform mentioned, including its communication with the danger alert system, is sketched in Figure 2.

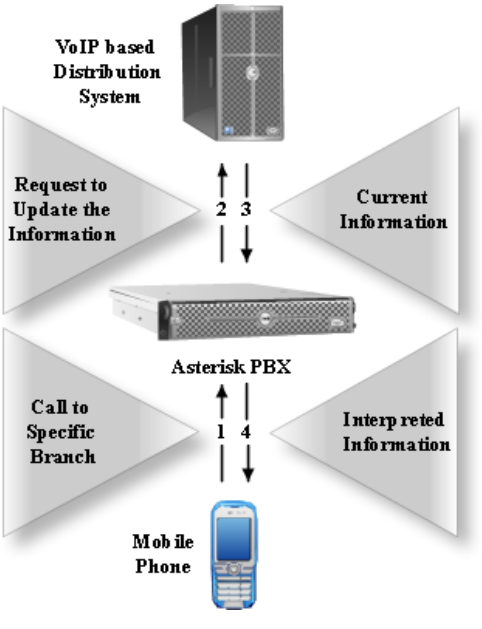

Fig. 2. Voice menu platform with the VoiceGlue system

As VoiceGlue requires a flawless and stable version of Asterisk to be able to function properly, the operating system's packages of Asterisk should be avoided and most recent version of the source code should be compiled and used. Before installing VoiceGlue, all dependencies for this application had to be installed first [3]. Once all dependencies have been installed, the VoiceGlue source code can be downloaded and the installation script launched. Subsequently, *dynlog, voiceglue* and *phoneglue* applications were installed. The first is a dynamic draw lots system, an analogy to Syslog. The second application serves as an interface between the system and VoiceGlue. *Phoneglue* then serves as an interface between the system, VoiceGlue and Asterisk.

3.1 Configuring Asterisk to Work with VoiceGlue

First, it was necessary to create a user 'phoneglue' to manage Asterisk's interface. This can be done in the */etc/asterisk/manager.conf* file by adding the following lines:

```
read = system,call,log,verbose,command,agent,user,
       originate
write = system,call,log,verbose,command,agent,user,
        originate
```

Subsequently, it was necessary to create a text string that will contact VoiceGlue via the Asterisk Gateway Interface (AGI). The basic functionality was achieved using the following dialplan in the Asterisk Extension Language (AEL). The dialplan below should be inserted into the */etc/asterisk/extensions.ael* file. Subsequently, Asterisk has to be restarted.

```
contextphoneglue {
    1 => {
Answer();
Agi(agi://158.196.244.148);
Hangup();
    };
};
```

3.2 Configuring VoiceGlue

Configuring VoiceGlue requires two steps. The first step is mandatory. A dialplan similar to that of Asterisk is implemented into the */etc/voiceglue.conf* file. This line should define which file will be interpreted at the VoiceGlue's launch. The following line has to be entered:

```
* http://158.196.244.148/katastrofix.vxml
```

This line tells the system to execute the *katastrofix.vxml* file whenever a call is placed on any extension. This file can be downloaded from the local web server through Hypertext Transfer Protocol (HTTP). Another possible configuration is to modify the Text-to-Speech module. This can be done in the */usr/bin/voiceglue_tts_gen* file. To use the Cepstral Swift voice generator, the following configuration was used:

```
#!/usr/bin/perl --      -*-CPerl-*-
$file = $::ARGV[2];
system ("/usr/local/bin/swift",  "-m", "text", "-o",
$file, $ARGV[1]);
```

When all changes have been made, you need to restart all three applications related to VoiceGlue. Applications should be shut in the following order: *VoiceGlue*, *Phoneglue* and *Dynlog*. Subsequently, the applications need to be launched again, this time in reverse order. To ensure that the system functions within a pre-defined system specification, it is also necessary to add a PHP script that retrieves the latest information from the communication system's web form and stores as a VXML document.

4 PHP Script for Conversion into VXML

This section describes the script which was used to build the interactive voice menu and which ensures the desired system functionality. The script is written in PHP. The script stores the information collected from the web form into the VoiceXML document format. The first part of the script is used to store the header of the output VXML file in the variable *vxmlstart* for the voice menu [4]:

```
<?php
$vxmlstart=
'<?xml version="1.0" encoding="UTF-8"\?\>
<vxml version = "2.0" xmlns="http://www.w3.org/2001/
vxml">
  <menu id="top" dtmf="true">
  <property name="inputmodes" value="dtmf"/>
  <prompt>
    <enumerate>
      For <value expr="_prompt"/>, press <value expr="
      _dtmf"/>
    </enumerate>
  </prompt>
';
```

The second part of the script is used to convert the information received through the web form in the VXML format. This is the information that you want the voice automat to play. Individual items for *menu* and *form* variables are created here:

```
echo "<html> <body>";
mysql_connect('localhost','voicexml', 'voice.*');
mysql_select_db('voicexml');
if($_POST)
{
  $choices=$_POST;
  for($i=1;$i<10;$i++)
  {
    if($choices[choice][$i])
```

```
    {
      $menu.='
        <choice next="#'.$i.'">
            '.$choices[choice][$i].'
        </choice>';
      $form.= '
        <form id="'.$i.'">
          <block>
            <prompt>
              '.$choices[text][$i].'
            </prompt>
            <goto next="#top" />
          </block>
        </form>
      ';
      $sql="UPDATE voicexml set choice='".$choices
      [choice][$i]."', text='".$choices[text][$i]."'
      where id=$i;";
      mysql_query($sql);
    }
  }
  $file=fopen("katastrofix.vxml", "w+");
  fwrite($file, $vxmlstart);
  fwrite($file, $menu."</menu>");
  fwrite($file, $form."</vxml>");
  fclose($file);
}
else
{
  $choices=mysql_query('select * from voicexml');
}
```

5 Conclusion

We developed an application in the VXML language that uses an interactive voice menu to obtain detailed information on an emergency. Using the VXML language, the Asterisk dialplan was outsourced into an external application – we opted for VoiceGlue. The system enables creating a voice menu without the need to record all voice messages in advance. This is ensured by incorporating the TTS module. The PHP script then ensures that the content collated through a web form is converted into VXML rather than HTML format.

Acknowledgments. This post leading to these results has received funding from G1 1485/2011 titled Modernization of laboratory exercises in the field of New Generation Information Systems in Switching course and from the Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 218086.

References

1. Voznak, M., Rezac, F., Zdralek, J.: Danger Alert Communication System. In: IWSSIP 2010 - 17th International Conference on Systems, Signals and Image Processing, Rio de Janeiro, Brazil, June 17-19 (2010) ISBN 978-85-228-0565-5
2. Meggelen, J.P., Smith, J., Madsen, L.: Asterisk: The Future of Telephony. O'Reilly Media, Sebastopol (2005)
3. VoiceGlue, http://www.voiceglue.org
4. McGlashan, S., Burnet, D., Carter, J., Danielsen, P., Ferrans, J., Hunt, A., Lucas, B., Porter, B., Rehor, K., Tryphonas, S.: Voice Extensible Markup Language (VoiceXML) Version 2.0, W3C - Recommendation (2004)
5. Bray, T., Paoli, J., Sperberg-McQeen, C.M., Maler, E., Yergeau, F.: Extensible Markup Language (XML) Fifth Edition, W3C - Technical Report (2008)
6. Lerdorf, R., Tatroe, K., MacIntyre, P.: Programming PHP, 2nd edn. O'Reilly Media, Sebastopol (2006)
7. Kwon, H.-J., Hong, K.-S.: A Design of User-Initiative Voice Web Using RSS and VoiceXML. In: 5th ACIS International Conference on Software Engineering Research, Management & Applications, SERA 2007, August 20-22, pp. 281–288 (2007)