

APPENDIX I

User Manual Of Our New Public-Domain SpeechWeb Browser

March, 2006

Supervisor: *Dr. Richard A. Frost*

Student: *Ma Xiaoli (William)*

Table of Contents

I. Installation Instructions

1.1 System Requirement

1.2 Install Opera 8 for Windows

1.3 Install Voice Feature for Opera 8

II. User Manual

2.1 Start the SpeechWeb Browser

2.2 Use the SpeechWeb Browser

2.3 List of Available Speech Input on Our Sample applications

I. Installation Instructions

1.1 System Requirement

This X+V voice browser needs an “Opera for Windows” version 8 or higher to be installed on the user’s windows system.

Recommended computer system requirements:

Operating System: Windows 2000 or XP.

CPU: Pentium 166MHz processor.

Memory: 32 MB of RAM.

Hard disk: 50 MB free disk space.

A speaker and a microphone, which are compatible with your computer system.

These system requirements may be updated without additional notification.

1.2 Install Opera 8 for Windows

Steps to install Opera 8 into your Windows system:

1. Go to the download page at <http://opera.com/download/>.
2. Select Opera 8 for Windows to download.
3. Double click on the Opera installation file you downloaded
4. The welcome screen tells you that: "You are about to install Opera".
Press "Next >"
5. Accept the software license agreement. Press "I Accept"
6. Install Opera in the suggested directory. Press "Next >"
7. Create icons and shortcuts. Press "Next >"
8. The information is complete and the installation will begin. Press
"Install" (Files are copied to your hard drive.)

9. Opera can start automatically after installation. Press "Finish"

If you have any trouble installing Opera 8 on your Windows system, please go to Opera Forum at <http://my.opera.com/community/forums/forum.dml?id=2>.

1.3 Install Voice Feature for Opera 8

Steps to install voice feature for Opera 8:

1. Download Opera 8 at <http://opera.com/download/>. Install Opera 8 for Windows into your computer.
2. Select "Tools" > "Preferences" > "Advanced" > "Voice". Enable the voice option.
3. After enabling the option, you will be asked to confirm that you want to install the voice libraries. Select confirm.
4. Voice is available after downloading Opera voice libraries.
5. To initiate a voice command, Press the "Voice" button on the View toolbar, or press the Scroll Lock key on your keyboard. Then say your command or query. After issuing, release the button.
6. You can also customize the "Voice key to talk", and "Talk key mode" in "Tools" > "Preferences" > "Advanced" > "Voice".

If you have any trouble to set up the voice feature into Opera, please go to Opera Voice Forum at <http://my.opera.com/community/forums/forum.dml?id=95>,

II. User Manual

2.1 Start the SpeechWeb Browser

Notice, Opera 8 for Windows with enabled voice feature has to be installed before using our X+V SpeechWeb Browser.

Steps to start the X+V SpeechWeb Browser:

1. Start your pre-installed Opera 8 for Windows. Make sure you have enabled voice feature.
2. Input the following URI into your Opera 8 web browser:
http://luna.cs.uwindsor.ca/~speechweb/p_d_speechweb/menu/demo_menu.xml Then Opera 8 will load the X+V SpeechWeb Browser Menu Page for you.
3. Make sure you have opened your sound speaker and set the volume high enough. You will hear a greeting voice message after you have successfully started the X+V SpeechWeb Browser.

Please note that SpeechWeb Browser also requires a microphone to allow the user to talk to the voice page.

2.2 Use the SpeechWeb Browser

In our SpeechWeb Browser, you can either talk to one of our sample voice pages (Sihlos), or ask SpeechWeb Browser to load your own voice page.

2.2.1 Talking to sample voice pages.

After successfully starting the SpeechWeb Browser, it loads the menu page of SpeechWeb Browser. You can choose one of the sample Sihlos to talk to. You may choose the Sihlo by clicking on their link, or simply saying it to menu

page(Please refers to Section 2.3 for the list of queries you can say in SpeechWeb Menu Page.) After you have chosen from one of the sample Sihlos, SpeechWeb will transfer you to that Sihlo, and you can start talking to it. You can also find the list of available speech input for Judy, Monty and Solarman in Section 2.3

Notice, you need to hold the “Voice” button on the Opera View toolbar or the Talk Key (‘Scroll Lock’ key by default) on your keyboard when you are speaking. Release the button after speaking.

2.2.2 Talking to your own SpeechWeb application.

If you have your own SpeechWeb application, you can input the URI of its X+V page into the text input-field, to load and talk to it. The text input-field is at the top of the SpeechWeb Menu Page and every sample Sihlos (*judy*, *monty*, *Solarman*). Please refer to the Developer’s Manual (in the Appendix II) for how to create your own SpeechWeb application in 4 easy steps.

2.3 Available Speech Input List

There is one menu page and three interpreters in the SpeechWeb Browser default sample application that you can talk to.

2.3.1 SpeechWeb Menu Page

In the menu page, you can say a choice from Judy, Monty, or Solarman to start talking. The possible query you can say is much more flexible.

For example:

“Can I talk to Judy”

“I wanna speak to Monty, please”

“Could you please transfer me to Solarman.”

“I would like to speak with Judy, please.”

“Please transfer to Monty, thanks”

“Judy, please”

There are more queries you can say. Simply try it by yourself. But, please note that, you have to include the name of the Sihlo you would like to talk to in your query, which means you have to say Judy, Monty, or Solarman in your query.

2.3.2 Judy

List of available speech input for Judy:

| *hello*

| *hello there*

| *hello judy*

| *goodbye*

| *goodbye judy*

| *fine thanks*

| *thanks*

| *thanks judy*

| *yes please*

| *what is your name*

| *who are you*

| *where do you live*

| *what do you know*

| *how old are you*

| *who made you*

| *what is your favorite band*

| *who is the vice president at the university of windsor*

| *who is the president at the university of windsor*

| *who is the president of sun microsystems canada*
| *who is the executive dean of science at the university of windsor*
| *who is the dean of science at the university of windsor*
| *tell me a poem*
| *know any poems*
| *tell me a joke*
| *know any jokes*
| *who is monty*
| *can I talk to monty*
| *can I talk to judy*
| *who is solar man*
| *can I talk to solar man;*

2.3.3 Monty

List of available speech input for Monty:

hello
| *hello there*
| *hello monty*
| *goodbye*
| *goodbye monty*
| *fine thanks*
| *thanks*
| *thanks monty*
| *yes please*
| *what is your name*
| *who are you*
| *where do you live*
| *what do you know*
| *how old are you*

| *who made you*
| *what is your favorite band*
| *who is the vice president at the university of windsor*
| *who is the president at the university of windsor*
| *who is the executive dean of science at the university of windsor*
| *who is the dean of science at the university of windsor*
| *tell me a poem*
| *know any poems*
| *tell me a joke*
| *know any jokes*
| *who is judy*
| *can I talk to judy*
| *who is solar man*
| *can I talk to monty*
| *can I talk to solar man;*

2.3.4 Solarman

Solarman can answer much more complicated question compare to Judy and Monty. You can ask question about the planets, the moons, and the people who discovered them in the solar system.

For example:

“How many moons orbit Mars”

“Which moons orbit Jupiter”

“Who discovered Titania”

APPENDIX II

Developer Manual Of Our New Public-Domain SpeechWeb Browser

May, 2006

Supervisor: *Dr. Richard A. Frost*

Student: *Ma Xiaoli (William)*

I. Components needed for your SpeechWeb application

1. Original knowledge source / program.
2. A CGI script.
3. A speech-grammar file.
4. A X+V page.

(Sometimes, we also refer to a SpeechWeb application as "*Sihlo*". Just make it different from other applications.)

III. How to create your own SpeechWeb application

The following is the four steps to create a SpeechWeb application:

1. Create a knowledge source / program.

You can use any programming language to create your knowledge source program. The program has only one requirement: it can receive a request and return a response as a string, no matter how the data processing is done inside the program. You can easily create it by modifying your conventional web application knowledge sources.

For example, the following is a small program written in Miranda, *judy.m*:

interpret "hello" = "hi there. My name is Judy"

interpret "hello there" = "Hello, how are you?"

interpret "hello judy" = "hi, how are you?"

interpret "goodbye" = interpret "goodbye judy"

interpret "goodbye judy" = "goodbye. Who do you want to talk to?"

interpret "fine thanks" = "Good, so am I. In fact I feel great."

interpret "thanks" = "no problem"

interpret "thanks judy" = "no problem at all"

interpret "yes please" = "yes please? What did you say? I was working on a new poem."

interpret "what is your name" = "My name is Judy."

interpret "who are you"

 = "My name is Judy. I know about poems."

interpret "where do you live"

 = "I live in a warm computer. "

 ++ "In good old Lambton Tower. University of Windsor."

interpret "what do you know"

 = "I only know one poem just now. My colleague, Dr. Frost"

 ++ " is looking for someone to teach me some more poems."

interpret "how old are you"

 = "What? That is a bit cheeky. I am younger than stuffy old Solar man."

interpret "who made you"

 = "I. B. M. and Opera Software made my ears and vocal chords. William Ma connected my "

 ++ "ears to my brain, and Doctor Frost, my colleague, made "

 ++ "my brain"

interpret "what is your favorite band"

 = "ARE. E. EM. They are a really cool band"

interpret "who is the vice president at the university of windsor"

 = "No idea, never been outside of Lambton Tower."

interpret "who is the president at the university of windsor"

 = "How should I know, this is the first time I have ever got to meet"

 ++ " anyone important."

interpret "who is the dean of science at the university of windsor"

 = "I know. Dr. Fryer. He helped get me more space for my memory."

```

interpret "tell me a poem"
  = "the dark clouds, lingered, as he lay."
  ++ " Then, the strong winds, blew them both away."

interpret "tell me a joke" = "do not know any jokes. But my friend, Monty, does"

interpret "know any jokes" = "No, but my friend, Monty does."

interpret "who is monty"
  = "Monty is my friend. He is a student"
  ++ " at the university of Windsor."

interpret "can I talk to judy"
  ="LINK=yes. here she is;"
  ++
  "SIHLO=http://sol.cs.uwindsor.ca/~speechweb/p_d_speechweb/judy/judy.xml;"

interpret "can I talk to monty"
  ="LINK=yes. here he is;"
  ++
  "SIHLO=http://sol.cs.uwindsor.ca/~speechweb/p_d_speechweb/monty/monty.xml;"

interpret "can I talk to solar man"
  ="LINK=yes. here he is;"
  ++
  "SIHLO=http://sol.cs.uwindsor.ca/~speechweb/p_d_speechweb/solarman/solarman.xml;"

interpret x = "sorry, got no poem for that one"

sh_answer x = interpret (drop 9 x)

```

2. Create a CGI script.

You will need to create a CGI script to handle the communication between your knowledge source program with the outside. This CGI script file needs to receive users' HTTP-request, and pass it to the program; also pass the returned response (from the program) back to the user.

For example, the following is the CGI script file (*judy.cgi*) for Judy:

```
#!/bin/csh -f

setenv HOME 'luna.cs.uwindsor.ca/fac3/richard/public_html/judy:$HOME'
setenv PATH 'luna.cs.uwindsor.ca/lapps1/mira:$PATH'

echo "Content-Type:text/plain"
echo "

setenv v "`luna.cs.uwindsor.ca/bin/cat`"
luna.cs.uwindsor.ca/lapps1/mira/bin/mira
luna.cs.uwindsor.ca//stu2/xing4/public_html/compile/judy.m << zzz
(sh_answer "$v")
/q
zzz
```

3. Create a JSGF speech-grammar file for your application.

You can find many tutorial about how to create a JSGF speech-grammar file from the Internet, here is one of them:

<http://java.sun.com/products/java-media/speech/forDevelopers/JSGF/>.

The easiest way to create this speech-grammar file is to simply list all of the possible input for your program. The following is the speech-grammar file used

for "*Judy Sihlo*":

```
grammar vxmlyjudy;  
  
public <s> = <simple>;  
  
<simple> = yes  
        | no  
        | go back  
        | hello  
        | hello there  
        | hello judy  
        | goodbye  
        | goodbye judy  
        | fine thanks  
        | thanks  
        | thanks judy  
        | yes please  
        | what is your name  
        | who are you  
        | where do you live  
        | what do you know  
        | how old are you  
        | who made you  
        | what is your favorite band  
        | who is the vice president at the university of windsor  
        | who is the president at the university of windsor  
        | who is the president of sun microsystems canada  
        | who is the executive dean of science at the university of windsor  
        | who is the dean of science at the university of windsor  
        | tell me a poem  
        | know any poems  
        | tell me a joke  
        | know any jokes  
        | who is monty  
        | can I talk to monty  
        | can I talk to judy  
        | who is solar man  
        | can I talk to solar man;
```

If you are a skillful speech application developer, then you can create a more complicated JSGP speech-grammar. The following is the speech-grammar file used in our SpeechWeb Browser menu page; it can recognize 549 different combinations of speech-input sentences.

```
grammar speechweb;
public <speechweb> = <start> <name> [please | thanks] {$= $name;};

    <start> = <sub_i> <action>
              | <sub_you> [please] <transfer_to>
              | NULL;

    <sub_i> = i wanna
              | i want to
              | i like to
              | i hope to
              | i would like to
              | can i
              | may i
              | shall i;

    <sub_you> = could you
               | would you
               | NULL;

    <action>= talk to | speak to | talk with | speak with | <transfer_to>;

    <transfer_to>= transfer [me] to;

    <name>   = judy | monty | solarman;
```

Here are some possible sentences that the menu page can recognize:

“Can I talk to Judy”

“I wanna speak to Monty, please”

“Could you please transfer me to Solarman.”

"I would like to speak with Judy, please."

"Please transfer to Monty, thanks"

"Judy, please"

4. Create the X+V page.

Before starting to create the X+V page, you need to have a copy of the identical X+V page, "*voicepage.xml*". It can be downloaded at http://luna.cs.uwindsor.ca/~speechweb/p_d_speechweb/voicepage.xml (Furthermore, you also need to know the URL of your CGI script and speech-grammar file.)

The following are the 3 changes that need to be done on your copy of "*voicepage.xml*":

1.) *Greeting message.*

At the beginning of the "*voicepage.xml*", you can find the following code:

```
<!-- Please modify the value of variable sv_greeting, and cgiLink to fit your application. -->
<script type="text/javascript">
/** The greeting message that will say to the user, only at the first time the user visits this
page. */
var sv_greeting="";
/** The link to your CGI interpreter location. Notice, you have to place the CGI interpreter
program with this page in the same domain to prevent a cross-domain security error.**/
var cgiLink="http://";
</script>
```

Change the value of the *sv_greeting* variable to your application greeting message. This message will be uttered automatically after the page is loaded. The following is the sample code on *Judy.xml* after the change:

```
<!-- Please modify the value of variable sv_greeting, and cgiLink to fit your application. -->
<script type="text/javascript">
/** The greeting message that will say to the user, only at the first time the user visits this
page. */
var sv_greeting="hi, my name is judy.";
/** The link to your CGI interpreter location. Notice, you have to put the CGI interpreter
program with this page in the same domain to prevent a cross-domain security error.*/
var cgiLink="http://";
</script>
```

2.) CGI script location.

Just next to the *sv_greeting* variable (the greeting message), you will find the variable called *cgiLink*. You should assign your CGI script URL to this variable.

The following is the *judy.xml* after changing the greeting message and the URL of its CGI script:

```
<!-- Please modify the value of variable sv_greeting, and cgiLink to fit your application. -->
<script type="text/javascript">
/** The greeting message that will say to the user, only at the first time the user visits this
page. */
var sv_greeting="hi, my name is judy.";
/** The link to your CGI interpreter location. Notice, you have to put the CGI interpreter
program with this page in the same domain to prevent a cross-domain security error.*/
var cgiLink="http://luna.cs.uwindsor.ca/~speechweb/p_d_speechweb/judy/judy.cgi";
</script>
```

3.) Speech-grammar file location.

There's only one dialog-field ('*vxml:field*') inside of VXML form ('*vxml:form*') in the *voicepage.xml* document. You only need to change the '*src*' attribute of the '*vxml:grammar*' element which is the first child element of '*vxml:field*'. You can easily find it at the beginning of the document and next to the *sv_greeting* and

cgiLink variables' declaration.

The following is the *voicepage.xml* before the change:

```
<!-- VoiceXML form. -->
<vxml:form id="vxml_form">
<vxml:field name="st_field" xv:id="voice_input" modal="true">
  <!-- NOTICE!!! PLEASE MODIFY THE VALUE OF 'src' ATTRIBUTE IN THE NEXT
LINE <grammar> ELEMENT TO YOUR GRAMMAR FILE LOCATION.-->
  <vxml:grammar type="application/x-jsgf"
    src="http://" />
```

The following is the code after the change for Judy, *judy.xml*:

```
<!-- VoiceXML form. -->
<vxml:form id="vxml_form">
<vxml:field name="st_field" xv:id="voice_input" modal="true">
  <!-- NOTICE!!! PLEASE MODIFY THE VALUE OF 'src' ATTRIBUTE IN THE NEXT
LINE <grammar> ELEMENT TO YOUR GRAMMAR FILE LOCATION.-->
  <vxml:grammar type="application/x-jsgf"
    src="http://luna.cs.uwindsor.ca/~speechweb/p_d_speechweb/judy/judy.jsgf/" />
```

III. Aware issues

Please also be aware of the following issues:

1. The knowledge source program should be an executable file, not a non-compiled source code file.
2. Make sure you give a read and execute permission to the public for your CGI script file. The UNIX command to change the permission is: *chmod* (e.g. *chmod 755 judy.cgi*)
3. Avoid using similar-sounding words to be the starting word of possible input choices in the knowledge-source program and the speech-grammar

file. For example, you can use "what time" and "which place" instead of "when" and "where".

4. You have to place your X+V page together with the knowledge-source program under the same Internet domain.
5. You can rename the X+V page "*voicepage.xml*" to any name you want but keep the extension to be ".xml".
6. If you have heard that "*there is an error in this application*" right after the browser is loaded, then it means you have given an incorrect speech-grammar file URL. Make sure you have "*http://*" at the beginning of the URL.
7. If a *SYSTEM ERROR* message appears after you have given your first question/request to the browser, then it could be caused by one or more of the following reasons:
 - a) No Internet connection. Please double-check your Internet connection, and make sure your firewall does NOT block it.
 - b) Invalid cross-domain connection [1]. To fix this problem you have to place your X+V page within the same Internet domain of your CGI script file. If you don't know what is meant by "same Internet domain", then simply place the X+V page and the CGI script file in the same folder/directory on your server.
 - c) Either an incorrect CGI script URL or the CGI script file in that URL does not exist. Double-check the spelling of the CGI script URL (*cgiLink*), and make sure it starts with a "*http://*".

[1], "About Cross-Frame Scripting and Security"

http://msdn.microsoft.com/workshop/author/om/xframe_scripting_security.asp

APPENDIX III New Public-Domain SpeechWeb Browser Website

We have created an Internet website for our new Public-Domain SpeechWeb Browser. It is the first place you should visit if you really want to try our SpeechWeb Browser. In there, you can find all the information about it, including the instruction to how to install it, links to sample SpeechWeb applications, and both User Manual and Developer Manual.

You are always welcome to visit our website, and give us any comments to our work. Our new Public-Domain SpeechWeb Browser website is at http://sol.cs.uwindsor.ca/~speechweb/p_d_speechweb/. You can use any regular web browser to open it, such as Microsoft IE, Firefox, Opera, etc.

The following is the URL to our SpeechWeb home page, which has links to both old and new version of our SpeechWeb Browser:
<http://sol.cs.uwindsor.ca/~speechweb/>.