RAPID VOICEXML DEVELOPMENT USING IBM'S GRAPHICAL CALL FLOW BUILDER

TR 29.3787

May 10, 2004

Ciprian Agapi
Felipe Gomez
James R. Lewis

IBM Pervasive Computing

Boca Raton, Florida

Abstract

The evolution of VoiceXML development can be traced from writing programs on text editors (like Notepad for example), followed by the emergence of IDE's that contained VXML editors, which gave immediate feedback on syntax errors. The next step in VoiceXML development has been the emergence of graphical VXML programming. Using intuitive shapes to create voice applications, IBM's graphical Call Flow builder greatly simplifies and speeds up the development of VXML prototypes.

ITIRC Keywords

VoiceXML
VXML
Call Flow
Graphical Development
Graphical VXML Development
Call Flow Builder
Voice Prototypes
Speech Recognition
Text To Speech

Table of Contents

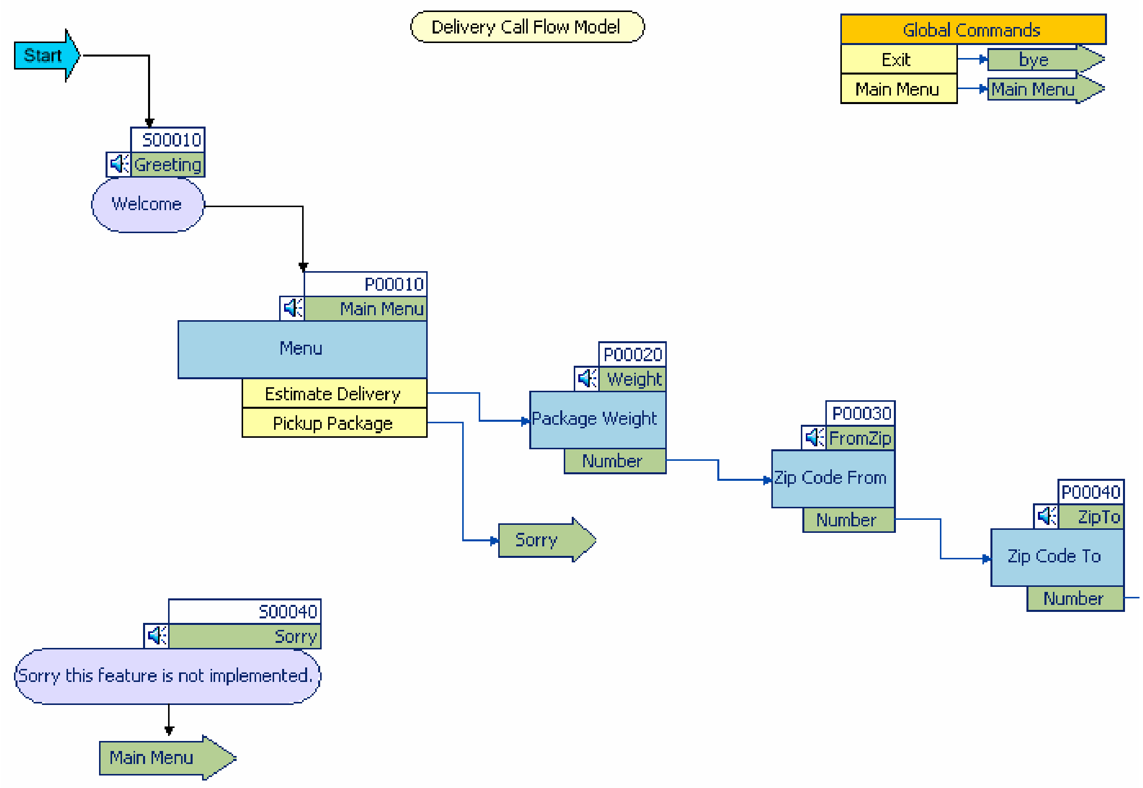
Introduction

IBM's Call Flow Builder is a very easy-to-use graphical tool to prototype and create voice applications. As with most tools, becoming familiar with it takes a little bit of time and effort, but in the long run, the time spent learning the tool will be greatly rewarded by exponential leaps in productivity.

Call Flow Layout

The Staircase Approach

When creating call flows, there are several ways to lay out the objects on the canvas. We recommend that users lay out the objects using a step-down approach, where objects are laid out towards the bottom right of the previous object. Using a consistent approach leads to more readable call flows. The following diagram illustrates this point:



Note that the objects are laid out in a staircase fashion. Also note that the GlobalCommands object appears on the top right hand corner of the first screen for easier readability and accessibility. The unimplemented feature object appears towards the bottom left hand corner of the screen in order to maximize the canvas real estate by clustering together functions and features used elsewhere in the call flow.

From General to Specific

To increase a user's productivity with the Call Flow Builder, we recommend that the development follows a top down work breakdown structure, and flows from the general to the specific. This way, a user can develop the voice application in several passes:

Step 1.

- In the first pass, a user identifies and creates the objects needed and the overall flow and connections of the application. When the objects are created, there is no need to worry about the exact wording of each object - this can be done in a later step. Instead, general names like "Greeting" or "Main Menu" can be used. For those objects that are being pointed to using GoTo's, it's advisable to enter a Name for easier back reference.

Step 2.

- In the second pass, a revision of each object can be made and values for the Optional fields in the property sheet can be entered. The optional fields contain the exact wording to be played back via TTS or Audio File. Also, in this pass DTMF support to User Responses (the yellow boxes) can be added by selecting the object and modifying its properties. At this stage, the desired level of support to the Catch properties can be added.

Step 3.

- In the third pass, a user should record the audio files using the List of Audio Files Required found in the .txt file by the same name as the .cfb file. Remember, DTMF audio files and Catch audio files need to be created from scratch using the names found in the List of Audio Files. Speech audio files can be easily created and recorded by simply clicking the audio icon in the object. In the future, the Sound Recorder will display the message that needs to be recorded. For now, the user might want to print the List of Audio Files and refer to it in order to record the correct message.

Step 4.

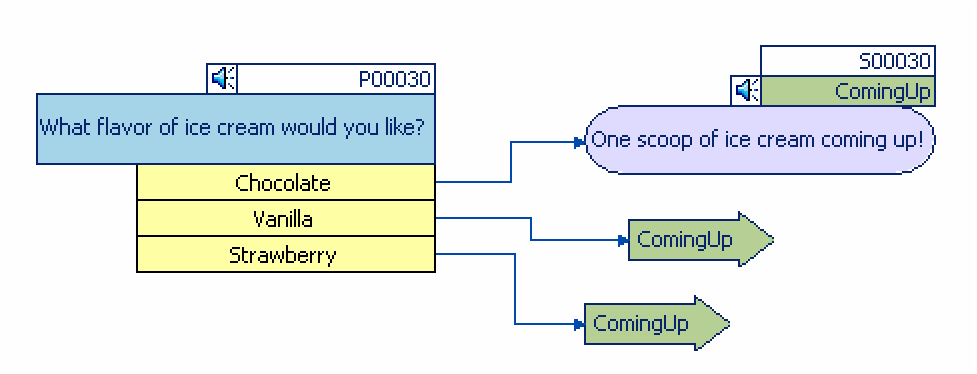
- In the fourth pass, the grammar files should be created. These can be associated with their Prompts by dragging and dropping the grammar file onto the figure in the canvas. Alternatively, a built-in grammar from the Select Grammar context menu in the Prompt properties field can be selected.

Step 5.

- In the fifth pass, the application can be tested by selecting “Simulate” from the context menu. This will help identify problem areas before running the VoiceXML Simulator. Also, he call flow can be validated selecting “Validate” from the context menu, and fixing any problem areas. Lastly, the VoiceXML code can be generated by selecting “Generate VoiceXML” from the context menu.

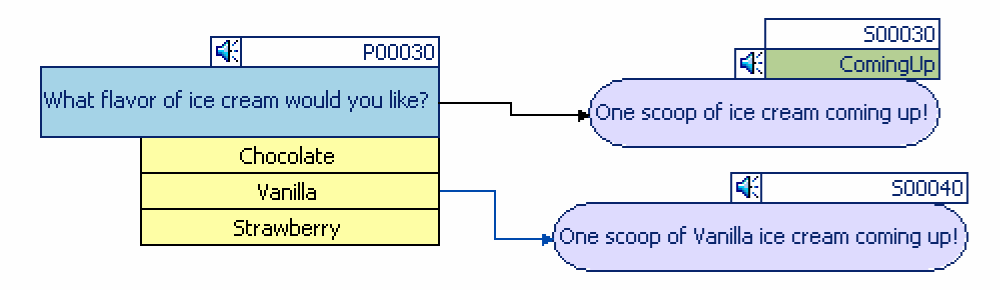
Connections

The current version of the Call Flow Builder, supports one -to-one connections only. That means that for every object on the call flow, there can only be one other object connected to it. In situations where several objects need to be connected to the same object, we recommend using GoTo's as in the following example:


P00030
What flavor of ice cream would you like?
Chocolate
Vanilla
Strawberry
S00030
ComingUp
One scoop of ice cream coming up!
ComingUp
ComingUp

In this case, saying "Vanilla" or "Strawberry" will redirect the call flow to the statement named **ComingUp**.

The previous example demonstrates the use of GoTo's as a means to support many-to-one connections. For that particular example, we could have used the Else-Path construct of the Prompt to achieve the same effect. The Else-Path construct refers to the default connection available from the centermost right or bottom points in the object. The following call flow illustrates this point:


P00030
What flavor of ice cream would you like?
Chocolate
Vanilla
Strawberry
S00030
ComingUp
One scoop of ice cream coming up!
S00040
One scoop of Vanilla ice cream coming up!

Saying "Vanilla" will redirect the cal**l** flow to the statement specific to Vanilla. Saying "Chocolate" or "Strawberry" will redirect the call flow to the default, else-path connection object, **ComingUp**.

Other Considerations

Support for DTMF

As an option to using Speech Recognition in the voice application, DTMF (Dual Tone Multi-Frequency) allows the use of touch tone keys of a telephone to respond to user prompts and selections. Adding DTMF support to the application is easy, and it guarantees a response from the user that is not subject to voice misrecos. To add DTMF to the speech application, a DTMF key to each of the User Responses can be assigned. Then, the Optional-Speech property for each of the Prompts should be modified, so that the wording accurately reflects the DTMF key assignments in addition to the Speech responses. For example: "Say or Press 1 for Chocolate, Say or Press 2 for Vanilla, etc." Adding DTMF later in the project will also require the re-recording of the speech audio files with the revised wording. Lastly, adding DTMF keys to the application can be especially useful when testing the voice application in the Simulator.

The Call Flow Builder supports three modes of interaction: Speech, DTMF and Speech-DTMF. By default, the code generated supports the Speech and Speech-DTMF mode. For users who wish to disable the speech recognition mode and support DTMF only, they can change the mode of the application by changing the following line in the .vxml file:

```
<var name="mode" expr="'speech'" />
```

to:

```
<var name="mode" expr="'dtmf'" />
```

Discussion

The VoiceXML code generated by the Call Flow Builder is sufficient enough to create and deploy simple voice applications, demos and prototypes. To create more complex, real world applications that include backend processing such as database access and other kinds of automation, a programmer still needs to be involved, to take code generated by the Call Flow Builder and modifying it to his/her needs. Additional programming skills required to deploy a full fledge application include Java, JSP, J2EE, and database programming, among others. In the future, end-to-end integration applications will further assist users in creating complex, real world voice applications.

The quality of a voice application depends entirely on the amount of planning and available resources for the task at hand. High-end, high-quality applications require many person hours to create, especially as it regards to organizing and deploying a professional solution that makes good use of specialized voice-talent, user interface designers, human factors engineers, programmers and project managers.

This document has been created to help users reach the top of their learning curve in the least amount of time. For questions, comments or feedback please visit the voice toolkit newsgroup at
http://www.alphaworks.ibm.com/forum/voicetoolkit.nsf/main

Conclusion

The Call Flow Builder is a rapid application development tool that facilitates the design and development of simple voice applications. It is our hope that this tool propels the expansion of the voice application market, much in the same way that graphical web page editors contributed to the explosive growth of the internet. Perhaps, one day, the voice application market will be as widespread as today's internet.

If there is one lesson to be learned from the growth of the internet, is that we need to take extreme care to create friendly and usable applications. In this way, our users will benefit greatly from them, and keep coming back for more. We encourage you to study the material in the following chapter, Recommended Reading Material.

We also encourage your questions, comments or feedback via the voice toolkit newsgroup at: http://www.alphaworks.ibm.com/forum/voicetoolkit.nsf/main

Recommended Reading Material

- Chapter 2: Designing a Speech User Interface, in the Voice Toolkit 5.0, VoiceXML Programmer's Guide, shipped with our Voice Toolkit. This describes IBM's recommended best UI practices, and is updated for each release of the toolkit.

- Balentine, B., & Morgan, D. M. (2001). How to build a speech recognition application: A style guide for telephony dialogs (2nd ed.). San Ramon, CA: Enterprise Integration Group. (http://www.amazon.com/exec/obidos/tg/detail/-/0967127823/qid=1082401580/sr=1-1/ref=sr_1_1/102-6721599-9080161?v=glance&s=books)

- Cohen, M. H., Giangola, J. P., and Balogh, J. (2004). Voice User Interface Design. Boston, MA: Addison-Wesley. (http://www.amazon.com/exec/obidos/tg/detail/-/0321185765/qid=1082401533/sr=1-1/ref=sr_1_1/102-6721599-9080161?v=glance&s=books)

Related Content

Voice Toolkit for WebSphere® Studio at:

http://www-306.ibm.com/software/pervasive/voice_toolkit/.

WebSphere Studio Trial Program at:
http://www-106.ibm.com/developerworks/websphere/downloads/WSsupport.html?S_TACT=103BGW01&S_CMP=campaign

Questions, comments or feedback at:
http://www.alphaworks.ibm.com/forum/voicetoolkit.nsf/main

Trademarks